An Optimization Framework to Enforce Multi-View Consistency for Texturing 3D Meshes

Zhengyi Zhao[1], Chen Song[2], Xiaodong Gu[1], Yuan Dong[1], Qi Zuo[1], Weihao Yuan[1], Liefeng Bo[1], Zilong Dong[1†], and Qixing Huang[2†]

[1] Institute for Intelligent Computing, Alibaba Group
[2] The University of Texas at Austin

Abstract. A fundamental problem in the texturing of 3D meshes using pre-trained text-to-image models is to ensure multi-view consistency. State-of-the-art approaches typically use diffusion models to aggregate multi-view inputs, where common issues are the blurriness caused by the averaging operation in the aggregation step or inconsistencies in local features. This paper introduces an optimization framework that proceeds in four stages to achieve multi-view consistency. Specifically, the first stage generates an over-complete set of 2D textures from a predefined set of viewpoints using an MV-consistent diffusion process. The second stage selects a subset of views that are mutually consistent while covering the underlying 3D model. We show how to achieve this goal by solving semidefinite programs. The third stage performs non-rigid alignment to align the selected views across overlapping regions. The fourth stage solves an MRF problem to associate each mesh face with a selected view. In particular, the third and fourth stages are iterated, with the cuts obtained in the fourth stage encouraging non-rigid alignment in the third stage to focus on regions close to the cuts. Experimental results show that our approach significantly outperforms baseline approaches both qualitatively and quantitatively. Project page: https://aigc3d.github.io/ConsistenTex.

Keywords: 3D texturing · Multi-view consistency · Optimization

1 Introduction

The performance of text-to-image models has reached a state where they can generate photorealistic 2D images of textured objects from a simple text prompt. This ability has stimulated a lot of interest in developing textured mesh models from a text prompt through multiple views of a 3D model. However, the idea of using 2D text-to-image models to synthesize 3D textures faces the fundamental challenge of achieving multi-view consistency among the 2D inputs.

There are two types of approaches towards achieving multi-view consistency. The first type [1, 2, 28, 33, 52] generates 3D textures purely based on 2D inputs and performs some type of averaging among overlapping regions of the 2D inputs. Although the goal is to retain high-quality details from the 2D inputs,

---

Fig. 1: Given a 3D mesh and a text prompt, we propose an optimization framework to synthesize the multi-view consistent texture. Top: meshes from a 3D generation model. Bottom: artist-created meshes.

the results among overlapping regions are usually blurry due to averaging. The second type [26, 38, 39, 45, 48–50] leverages 3D data to enhance multi-view consistency, e.g., by using attention modules trained from rendered images of 3D models. These methods improved multi-view consistency, but the results tend to overfit rendered images and lose realistic image details in synthetic 2D images.

This paper presents a novel text-to-textured-mesh approach that achieves multi-view consistent results. Our approach makes contributions in both types of methods and combines their strength in a seamless way. While our approach also employs 3D data for training, we focus on promoting multi-view consistency in global style and shape that possess a minimal domain gap between rendered images and synthetic 2D images. On the other hand, we allow for global drifts in image details, which are the main source of overfitting. To achieve multi-view consistency, we perform non-rigid alignments among overlapping regions of synthetic 2D images. Non-rigid alignments are performed among a subset of images selected from an over-complete of synthetic 2D images.

Specifically, our approach proceeds in four stages. The first stage uses a pre-trained text-to-image model to generate an overcomplete set of candidate textures from a predefined set of viewpoints. The second stage optimizes a subset of views that are mutually consistent. The third stage performs joint alignment among selected 2D textures to maximize consistency across overlapping regions. The fourth stage cuts and stitches the aligned texture images into a consistently textured mesh. The third and fourth stages are iterated in an alternating manner, where the cuts obtained in the fourth stage guide the non-rigid alignment procedure so that alignment focuses on regions near the optimized cuts.

Experimental results show that our approach outperforms state-of-the-art text-to-texture approaches both qualitatively and quantitatively. Qualitatively, our approach is much more photorealistic and does not present discontinuous or blurry texture elements that are frequent among other approaches. Quantitatively, our approach achieves a better FID score than the existing methods.

2   Related Work

Multi-View Representation. Multi-view representations are used in many 3D recognition approaches [13, 14, 16, 21, 42, 47], where the key idea is to take advantage of large-scale pre-trained 2D networks. Our approach is particularly related to methods that perform 3D object reconstruction by predicting depth images from given viewpoints [40, 44]. However, these approaches simply fuse the multi-view predictions and fail to enforce the multi-view consistency. Therefore, the reconstruction is blurry and unrealistic.

Consistent Multi-View Generation. A challenge in text-to-textured-mesh models is ensuring the multi-view consistency among the text-to-image outputs. Dreamfusion [31] optimizes the geometry and color of a NeRF through SDS loss. Zero123 [25] successfully employs a camera transformation as condition to synthesize novel views from an image. TextMesh [45] leverages quartet canonical views for mesh rendering, yet it tent to yield blurry results. Several methods [26, 38, 39, 48–50, 55] are trained on 3D rendered datasets [7] to enhance consistency but tend to lose realistic details in generated images. Moreover, most of them are agnostic to depth condition, which is crucial for alignment between 2D images and 3D meshes.

Text-to-Texture Models. Several pioneers [4, 29, 30] are trained on a semantic CLIP loss, but still struggle to generate high quality textures with text condition. Trained with a large dataset [35], diffusion models [34] can generate high-quality images with rich details from text prompts. On this basis, additional conditions are added to provide 3D structure-aware through rendering of meshes, such as depth maps and inpainting masks [54]. Some methods [2, 33] generate single view 2D images on these conditions and incrementally merge them into one mesh texture. However, these approaches can easily lead to suboptimal results due to accumulated errors and the lack of global perception. Using a latent UV map [1,12,19,27,28] as a unified representation has demonstrated its effectiveness of increasing global consistency. Since latent pixels compress a patch in RGB images, the decoded RGB pixels cannot guarantee 3D consistency, which leads to the blurriness or inconsistent details in the textures. The other methods [52, 53] finetune a diffusion model on latent UV space to avoid projection, but also generate artifacts with poor UV expansions. Besides, generating PRB materials is also considered as a focus in some work [3, 8, 51]. In contrast, our approach performs alternating optimization of non-rigid alignment among the selected images, as well as joint optimization of the best cuts to guide the non-rigid alignment.

View Selection. The quality of a multi-view representation is highly dependent on the distribution of source 2D views. View selection is related to the view saliency problem, which has been extensively studied in the existing literature [5, 10, 18, 22, 23, 36, 41]. Closely related to our work, Sun *et al.* [43] present an approach to optimize camera poses to maximize the coverage of an input scene. Wei *et al.* [47] optimize viewpoints to maximize the quality of fusion of the learned 2D features. We introduce a view selection approach that selects a subset of consistent views. Driven by the goal of enforcing multi-view consistency, we

propose a novel formulation that optimizes a view-consistent score among the chosen images, while constraining that the selected images provide complete coverage of the underlying 3D model.

Texture Cutting and Stitching. Cutting and stitching overlapping images is a fundamental approach to texture synthesis. The pioneering work of Efros and Freeman [11] optimizes cuts between pairs of overlapping patches for texture synthesis. Our method formulates cut optimization and texture stitching as a second-order MRF problem on the mesh faces, where the second-order term models view consistency across different images. This MRF formulation is introduced by Waechter *et al.* [46], and our contribution lies in the jointly aligned images as input. In particular, we find it essential to alternate between solving the MRF problem and using the MRF output to refine its inputs.

3   Overview

3.1   Problem Statement

Suppose that we have a 3D model represented as a triangular mesh $\mathcal{M} = (\mathcal{V}, \mathcal{F})$ where $\mathcal{V}$ and $\mathcal{F}$ denote the vertex and face sets, respectively. Given a text prompt as input, our goal is to convert $\mathcal{M}$ into a textured mesh.

3.2   Approach Overview

View generation. The first stage takes an overcomplete set of n views of the 3D model $\mathcal{M}$ and the associated depth images as input and outputs the corresponding RGB-D images using a text-to-image model. Consistent with TexFusion [1], we exploit a depth conditioning image generation model. At this stage, the output RGB-D images are generated in isolation, and inconsistencies across views are expected.

View selection. The second stage selects a subset of RGB-D images that are consistent with each other. This is formulated as a solution to a quadratic assignment problem in which the objective function models the consistency scores between pairs of RGB-D images, under the constraint that the selected images provide a complete coverage of $\mathcal{M}$. We introduce a novel sequential semidefinite programming approach to solve this problem accurately and efficiently.

View alignment. The third stage performs joint non-rigid warping of the selected images so that the pixel colors after warping agree with each other. We first solve a joint optimization problem to alleviate illumination changes in the selected images. We then use SIFTFlow [24] to precompute dense correspondences between overlapping views. The resulting correspondences are fed into another joint optimization problem to optimize the deformation of each view so that the deformed images are view-consistent.

View cutting and stitching. The final stage outputs a cut of each aligned image so that they form a seamless textured output. We formulate this as solving a second-order MRF problem to optimize the label of each mesh face, where the
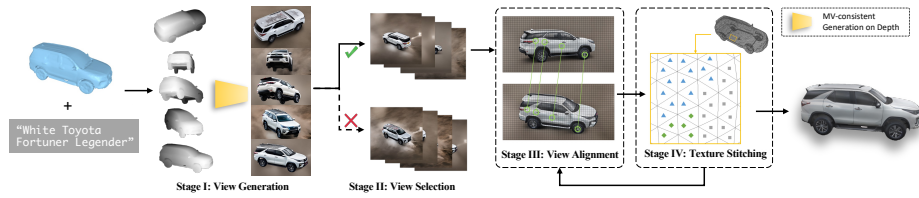
Fig. 2: The overall pipeline of our approach, which consists of four stages. The first stage (view generation) uses a pre-trained text-to-image model to generate an over-complete set of RGB-D images of the input model. The second stage (view selection) selects a subset of consistent RGB-D images from the view generation output. The third stage (view alignment) performs non-rigid warping to further improve the multi-view consistency among the selected images. The last stage (texture stitching) produces cuts between pairs of overlapping images by solving a second-order MRF problem.

pairwise terms model view consistency. The output is a textured mesh with aligned images as the texture source.

The output provides cuts between pairs of overlapping images along which the pixel colors of the deformed images agree. We then use these cuts to guide the view alignment to focus on regions near these cuts to make the input to the MRF more consistent. Empirically, we find 2 to 3 iterations of view alignment and view cutting and stitching sufficient.

4   Approach

As illustrated in Fig. 2, our approach consists of four stages. We first describe view generation and view selection in Sec. 4.1 and Sec. 4.2, respectively. We then elaborate on view alignment in Sec. 4.3. Finally, we introduce texture cutting and stitching and discuss how to alternate between this stage and view alignment in Sec. 4.4.

4.1   Stage I: View Generation

The purpose of this stage is to generate initial multi-view images from the input text prompt. Directly generating high-quality images that are also consistent using state-of-the-art diffusion-based techniques is extremely difficult. Existing approaches typically explore a trade-off between single-image quality and multi-view consistency. The difference in our approach is that we have a post-processing step that combines view selection, view alignment, and view stitching. This offers us freedom to design the view generation procedure to maximize the performance of the final output.

Specifically, our post-processing step allows drifted image details as it performs alignment. However, it cannot create image details. Therefore, the multi-view outputs should contain texture details of the final output. Moreover, view alignment is most effective when the absolute displacements are local, meaning
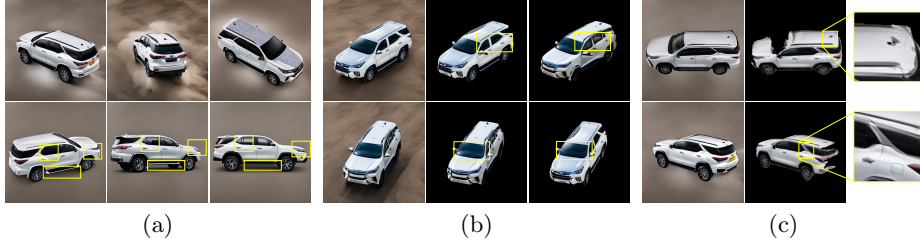
Fig. 3: Three types of multi-view inconsistencies from pre-trained text-to-image models. (a) The appearance of the image content may change significantly, even with slight perturbations in the camera pose (see the bottom row). (b) There are variations in illumination among overlapping regions of different views. (c) There are drifts in the detail of the image between overlapping regions of different views. (The left column images in (b) and (c) are warped with each other.)

some global consistency among the output multi-views is also important. To this end, we introduce a three-phase diffusion procedure to generate the multi-view outputs.

Similar to Zero123++ [38], the first phase leverages 3D priors to guide the diffusion procedure. We first utilize a multi-view consistent model (see supp. material) finetuned on the GObjaverse [32] dataset to provide 3D prior. Note that this model is applied only at the initial diffusion step 800. It ensures the global consistency of the image outputs, yet it has minimal influence on later diffusion steps that are responsible for recovering image details.

The second phase, which runs between diffusion steps 800-200, adopts latent UV mapping [28], a popular data-free approach to enforce multi-view consistency. The key idea is to aggregate multiple views in a shared latent map that is then mapped back to individual views. Although improved consistency is achieved, image details can still be dropped during the aggregation step, which performs some type of averaging (see supp. material). Therefore, similar to SyncMVD [27], we do not perform latent UV mapping to generate the final output.

The final phase performs diffusion on each view independently to generate texture details.

As shown in Fig. 3, the multi-view image generation results are consistent in global style and shape. However, local shapes and styles may differ and there are also drifts. Another important idea is to generate an over-complete set of images and operate on a subset of more consistent images to generate the output. This approach nicely mitigates inconsistency introduced in the third phase, which is performed on each view independently.

4.2 Stage II: View Selection

With $\mathcal{I} = \{I_1, \cdots, I_n\}$, we denote the result of stage I. The goal of the second stage is to select a subset of images $\mathcal{I}^\star \subset \mathcal{I}$ such that 1) the quality of each

image is high, 2) the images are mutually consistent, and 3) the images cover the underlying 3D object. In the following, we first introduce a general optimization formulation that takes as input a single image scoring function $s(I_i)$ and a consistency scoring function $s(I_i, I_j)$. We then discuss how to define $s(I_i)$ and $s(I_i, I_j)$ in our setting.

Optimization formulation. Given $s(I_i)$ and $s(I_i, I_j)$, we solve a constrained optimization problem to select consistent images. We introduce an indicator variable $z_i \in \{0, 1\}$ for each I_i, where $z_i = 1$ indicates that I_i is selected and vice versa. Let vector $\boldsymbol{z}$ collect all indicators, while vector $\boldsymbol{s} = (s(I_i)) \in \mathbb{R}^n$ and matrix $S = (s(I_i, I_j)) \in \mathbb{R}^{n \times n}$ collect image quality and consistency scores.

Our goal is to optimize $\boldsymbol{z}$ to minimize $\boldsymbol{s}^T\boldsymbol{z} + \lambda\boldsymbol{z}^T S\boldsymbol{z}$ while ensuring that the selected images cover the underlying model. To this end, we model the coverage constraint as $C\boldsymbol{z} \geq \boldsymbol{1}$, that is, each row of C constrains that one face on the underlying mesh is covered by at least one selected image. This leads to the following optimization problem:

$$\underset{\boldsymbol{z} \in \{0,1\}^n}{\text{minimize}} \quad \boldsymbol{s}^T\boldsymbol{z} + \lambda\boldsymbol{z}^T S\boldsymbol{z} \qquad \text{subject to} \quad C\boldsymbol{z} \geq \boldsymbol{1}. \tag{1}$$

Equation (1) is hard to optimize, due to non-convexity and integer variables. We introduce two relaxations. The first relaxes z_i so that they can take real values between 0 and 1, and the second uses the semidefinite programming relaxation to $\boldsymbol{z}^T S\boldsymbol{z}$. More precisely, we introduce an additional matrix variable $Y = \boldsymbol{z}\boldsymbol{z}^T$ and apply the standard relaxation approach as $Y \succeq \boldsymbol{z}\boldsymbol{z}^T$ and $\text{diag}(Y) = \boldsymbol{z}$. Furthermore, we maintain an index set $\mathcal{I}_f$ that specifies all currently selected images. Initially, we $\mathcal{I}_f = \emptyset$. This leads to the following semidefinite program:

$$\underset{\boldsymbol{z} \in \mathbb{R}^n, Y \in \mathbb{R}^{n \times n}}{\text{minimize}} \quad \langle \boldsymbol{s}, \boldsymbol{z} \rangle + \lambda \langle S, Y \rangle$$

$$\text{subject to} \quad C\boldsymbol{z} \geq \boldsymbol{1}, \quad 0 \leq \boldsymbol{z} \leq 1, \quad 0 \leq Y, \quad \text{diag}(Y) = \boldsymbol{z},$$

$$z_i = 1, \quad \forall i \in \mathcal{I}_f, \quad \begin{pmatrix} Y & \boldsymbol{z} \\ \boldsymbol{z}^T & 1 \end{pmatrix} \succeq 0, \tag{2}$$

where $Y \succeq \boldsymbol{z}\boldsymbol{z}^T$ is identical to Eq. (2). We then augment $\mathcal{I}_f$ with $i^\star = \text{argmax}_{i \notin \mathcal{I}_f} z_i$. This procedure is iterated until $C\boldsymbol{z} \geq 1$ is satisfied. For simplicity, we assume that the selected views are the first $k < n$ images $I_i, 1 \leq i \leq k$.

Image score $s(I)$**.** Our key observation is that if the quality of an image I is high, then the pixel colors tend to repeat among other input images. This motivates us to build a probabilistic color model for each vertex $v \in \mathcal{V}$ (cf., Fig. 4 (Left)). Specifically, denote $\mathcal{I}_v \subset \mathcal{I}$ as the subset of images in which v is visible. Let $\boldsymbol{c}(v, I)$ be the color of v in I. We first perform mean shift clustering [6] to extract the largest cluster of $\{\boldsymbol{c}(v, I)|I \in \mathcal{I}_v\}$, which represents the most represented color. With $\boldsymbol{c}_v$ and σ_v we denote the center and variance of this cluster. We then define

$$p(v, \boldsymbol{c}) = \frac{1}{(2\pi)^{\frac{3}{2}}} \frac{1}{\sigma_v^3} \exp\left(-\frac{\|\boldsymbol{c} - \boldsymbol{c}_v\|^2}{2\sigma_v^2}\right). \tag{3}$$
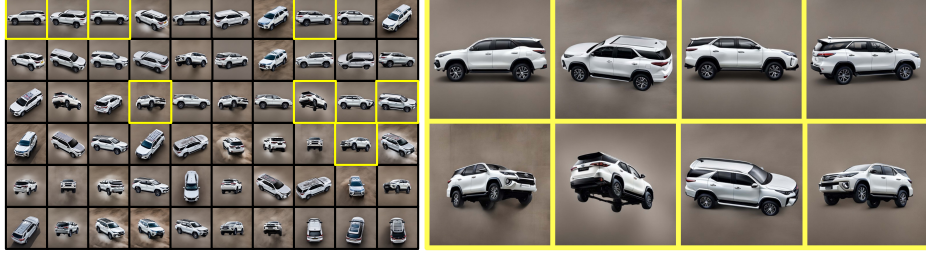
Fig. 4: Illustration of view selection. (Left) The input images ranked in the increasing order of $S(I_i)$. (Right) The selected images that consider the image scores $S(I_i)$, the consistency scores $S(I_i, I_j)$ and the covering constraint.

Given an image I, let $\mathcal{V}_I \subset \mathcal{V}$ collect all visible vertices in I. We sort the values $\{p(v, \boldsymbol{c}(v, I)) | v \in \mathcal{V}_I\}$ in increasing order and set the score $s'(I)$ of I as the $t\%$ percentile ($t = 20$) of $p(v, \boldsymbol{c}(v, I))$. The final score is given by $s(I) = (s_{\max} - s'(I))/(s_{\max} - s_{\min})$ where $s_{\max}(s_{\min}) = \max_I(\min_I)s'(I)$. As shown in Fig. 4 (Left), high-quality images tend to have small values in $s(I)$.

Consistency score $s(I_i, I_j)$. For the purpose of view selection, we find that defining $s(I_i, I_j)$ based on mean color differences without alignments is sufficient.

We set $s(I_i, I_j) = 0$ if I_i and I_j do not overlap. Figure 4 (Right) shows that the consistency score is critical to selecting consistent images.
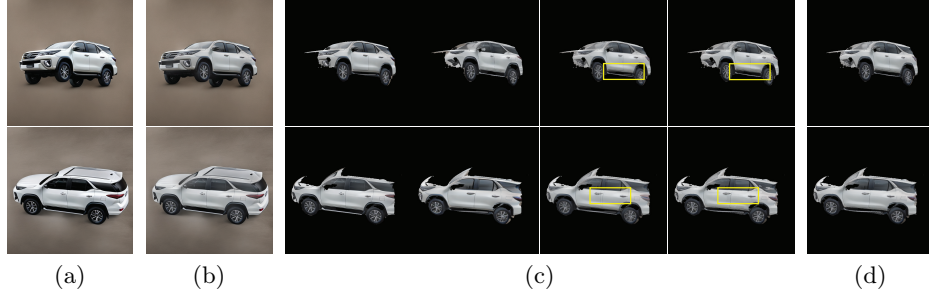
4.3   Stage III: View Alignment

The third stage of our approach performs non-rigid alignment among the selected RGB-D images, making them consistent among their overlapping regions. As illustrated by Fig. 5, we achieve this goal in two steps. The first step adjusts the pixel colors together to reduce the illumination variations in the selected images. The second step performs joint image warping to align image features across the selected images. In both steps, we parameterize the deformation of each image using the free-form deformation (FFD) [37] on a regular $L \times L$ lattice (we choose $L = 20$ for color adjustment and $L = 40$ for image warping). Note that it is possible to avoid illumination variations by training text-to-image models from rendered 3D shapes. However, our goal is to use photorealistic images from pre-trained text-to-image models. Therefore, we chose to solve an optimization problem to consistently reduce color variations.

Color adjustment. With $I_i^x \in \mathbb{R}^{m \times m}$, where $x \in \{r, g, b\}$ denote the color channel the image I_i, we parameterize the adjusted image $\hat{I}_i^x$ as:

$$\hat{I}_i^x(\boldsymbol{p}) = B(\boldsymbol{p})\boldsymbol{c}_{i,x}^1 + I_i^x(\boldsymbol{p})B(\boldsymbol{p})\big(1 - \boldsymbol{c}_{i,x}^0 - \boldsymbol{c}_{i,x}^1\big), \tag{4}$$

where $B(\boldsymbol{p}) \in \mathbb{R}^{1 \times L^2}$ is a sparse vector that encodes the FFD coefficients of pixel $\boldsymbol{p}$; $\boldsymbol{c}_{i,x}^0$ and $\boldsymbol{c}_{i,x}^1$ are control weights that correspond to all black and white pixels. Let M_i be the diagonal matrix that specifies the inactive control weights

(a)   (b)   (c)   (d)

Fig. 5: Illustration of the view alignment stage. We show the result of two overlapping images through the joint view alignment procedure. (a) Two input images. (b) After adjustment of the global illumination. (c) From left to right: overlapping regions of each input image, warped images using ground-truth maps between the overlapping regions, overlaid results of the first two columns, and overlaid after SIFTFlow alignment. (d) Overlaid alignments after joint alignment. The pairwise SIFTFlow alignments are preserved.

(*i.e.*, those that correspond to the background pixels). The control weights are restricted by $0 \leq c_{i,x}^0$, $0 \leq c_{i,x}^1$, $c_{i,x}^0 + c_{i,x}^1 \leq 1$, and $M_i c_{i,x}^0 = M_i c_{i,x}^1 = 0$, which ensure that $\hat{I}_i^x \in [0,1]^{m \times m}$.

We use $(I_{ij} \subset I_i, I_{ji} \subset I_j)$ to denote the regions that overlap between I_i and I_j. Let $g_{ij} : I_{ji} \to I_{ij}$ and $g_{ji} : I_{ij} \to I_{ji}$ be the ground-truth correspondence maps between them. We solve the following quadratic program with linear constraints to determine $c_{i,x}^0$ and $c_{i,x}^1$:

$$\min_{\{c_{i,x}^l\}} \sum_{x \in \{r,g,b\}} \left(\sum_{i \neq j} \sum_{p \in I_{ij}} \|\hat{I}_i^x(p) - \hat{I}_j^x(g_{ij}(p))\|^2 + \mu \sum_{i=1}^k \sum_{l=0}^1 {c_{i,x}^l}^T L c_{i,x}^l \right)$$
$$s.t. \quad 0 \leq c_{i,x}^0, 0 \leq c_{i,x}^1, c_{i,x}^0 + c_{i,x}^1 \leq 1, \quad M_i c_{i,x}^0 = M_i c_{i,x}^1 = 0, \tag{5}$$

where L is a Laplacian matrix of the control lattice. Figure 5 (b) shows that this step can effectively reduce the variation of illumination between overlapping images.

Image warping. Let $(O_{ij} \subset I_{ij}, O_{ji} \subset I_{ji})$ be the focused overlapping subsets between I_i and I_j. Our goal is to deform each I_i so that O_{ij} and O_{ji} are aligned among the deformed images. Initially, $O_{ij} = I_{ij}$ and $O_{ji} = I_{ji}$. After determining the optimal cut between I_i and I_j, we recompute O_{ij} and O_{ji} as the dilated regions along the cuts. This allows us to focus on alignments among potentially more consistent regions to improve alignment results.

Let $f_{ij} : g_{ij}(O_{ji}) \to O_{ij}$ and $f_{ji} : g_{ji}(O_{ij}) \to O_{ji}$ be dense pixel-wise alignments between the focused overlapping regions under the view of each image. In our experiments, we compute f_{ij} and f_{ji} using SIFTFlow [24], a state-of-the-art method that establishes high-fidelity dense correspondences (Fig. 5 (c)).

Let $c_i \in \mathbb{R}^{2 \times L^2}$ be the grid of control points that determine the deformation $\mathcal{D}_i$ of the image I_i. The deformed position of p is given by $\mathcal{D}_i(p) = c_i b(p)$.

Let $\boldsymbol{c}_i^{(0)}$ be the initial control grid of $\mathcal{D}_i$, we model the following optimization problem to optimize $\{\boldsymbol{c}_i\}$ jointly:

$$\min_{\{\boldsymbol{c}_i\}} \sum_{i \neq j} \sum_{\boldsymbol{p} \in O_{ij}} \|\boldsymbol{c}_i \boldsymbol{b}(\boldsymbol{p}) - g_{ij}(\boldsymbol{c}_j \boldsymbol{b}(g_{ji}(\boldsymbol{p}))) - f_{ij}(\boldsymbol{p})\|^2$$

$$+ \sum_{i=1}^{k} \left(\mu_1 \|\boldsymbol{c}_i - \boldsymbol{c}_i^{(0)}\|^2 + \mu_2 \mathrm{Tr}(\boldsymbol{c}_i L \boldsymbol{c}_i^T) \right), \tag{6}$$

where $\mu_1 = 2$ and $\mu_2 = 200$ in all our experiments. Since Eq. (6) consists of non-linear least squares, we apply the Gauss-Newton method for optimization.

Figure 5 (d) shows that joint view alignment offers deformations of each image that preserve the alignments provided by pairwise SIFT flows.

4.4   Stage IV: Texture Stitching

The final stage of our approach takes the aligned RGB-D images $I_i^\star = \mathcal{D}_i^\star(I_i), 1 \leq i \leq k$ as input and outputs a complete textured 3D model. This is formulated as decomposing the mesh faces $\mathcal{F}$ of the input mesh $\mathcal{M}$ into sets k: $\mathcal{F} = \mathcal{F}_1 \cup \cdots \mathcal{F}_k$ where $\mathcal{F}_i$ is associated with $I_i^\star$. We formulate this procedure as a joint labeling problem. Specifically, we associate each face $f \in \mathcal{F}$ with a random variable $z_f \in \{1, \cdots, k\}$ that specifies the set of faces $\mathcal{F}_{z_f}$ to which it belongs. Let $\mathcal{E} = \{(f, f')\}$ collect all pairs of adjacent faces. We optimize z_f by minimizing the following objective function:

Fig. 6: (Left) Optimized cut among two overlapping images showing mis-alignment along the seams of different images. (Right) Overlaid SIFTFlow alignment results among the masked regions, where the seams are aligned.

$$\min_{\{z_f\}} \sum_{f \in \mathcal{F}} \phi_f(z_f) + \mu \sum_{(f,f') \in \mathcal{E}} \phi_{ff'}(z_f, z_{f'}) \tag{7}$$
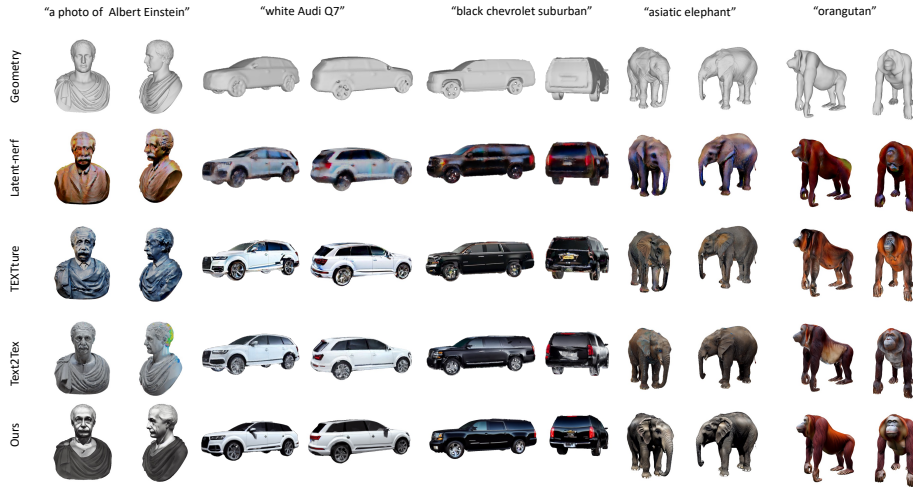
where $\phi_f(z_f)$ is the cost of label z_f, and $\phi_{ff'}(z_f, z_{f'})$ is the consistency between z_f and $z_{f'}$. In this paper, we define $\phi_f(z_f)$ using a variant of Eq. (3), that is, a label has a small cost if it is more consistent with other images. Furthermore, $\phi_{ff'}(z_f, z_{f'}) = 0$ if $z_f = z_{f'}$. Otherwise, its value depends on the color difference between z_f and $z_{f'}$. Details are deferred to the supp. material.

We solve Eq. (7) using the OPENGM2 [17] implementation of TRWS [20]. Figure 6 (Left) shows an example of optimized cuts between aligned images.

After obtaining $\mathcal{F}_i, 1 \leq i \leq k$, we use the corresponding cut between each pair of overlapping images I_i and I_j to recompute O_{ij} and O_{ji}. This is achieved by dilating the cut in each I_i by l pixels ($l = 20$) and taking the union of the dilated regions to define O_{ij} and O_{ji}. After that, we rerun the view alignment to

Fig. 7: Qualitative visualization. From top to bottom: Input 3D mesh; Latent-NeRF; TEXTure; Text2Tex; Ours.

generate the inputs for texture stitching. This alternating procedure is iterated until the final output becomes steady (typically 2-3 iterations). Figure 6 (Right) shows that this alternate procedure can enhance visual consistency along the cuts.

5   Experimental Results

We begin with the experimental setup in Sec. 5.1. We then present the experimental results in Sec. 5.2. Finally, we describe an ablation study in Sec. 5.3.

5.1   Experimental Setup

Dataset. We collect 300 models from different categories in Objaverse [7]. We also collect 20 models used in other text-to-texture-mesh methods and models from generation approaches [9, 32] for qualitative comparison.

Baseline approaches. We evaluate our method against Latent-NeRF [28], TEXTure [33], Text2Tex [2], SyncMVD [27], and Zero123++ [38] the state-of-the-art text-to-textured-mesh approaches with open source implementations. We also compare with a popular multi-view texturing method (MVS-Texturing) [46] to show the superiority of our global optimization framework.

Quantitative Comparisons. Qualitative evaluation of generative models is always a difficult problem. We follow TexFusion [1] and use the same qualitative evaluation metric (*i.e.*, using the generated image based on the depth condition as an upper bound to calculate the FID [15]). In addition, a user study is designed in which human participants are asked to assign a numerical score between 1 and 10 for the generated result.

Fig. 8: Global consistency and local consistency comparisons between SyncMVD and Ours.

5.2   Analysis of Results

Figure 7 compares our approach to the baseline approaches in five models with text prompts. Visually, our approach preserves the appearance among the individual images while ensuring a globally consistent texture output. In contrast, the results of Latent-NeRF are blurry. This is due to the averaging operation performed on the inconsistent outputs from pretrained text-to-image models. TEX-Ture and Text2Tex demonstrate some qualitative improvement over Latent-NeRF. However, the results of TEX-Ture and Text2Tex still exhibit noticeable visual artifacts due to accumulated errors in the sequential procedure.

Fig. 9: The left part: comparison between Zero123++, MVS-Texturing and Ours. The right part: intermediate results of Zero123++.

Figure 8 compares our approach to SyncMVD [27]. Our approach reduce some of the global inconsistencies due to the 3D prior. The details are also clearer thanks to our optimization in rgb space.

Figure 9 compares our approach to Zero123++ [38] and MVS-Texturing [46]. We generate the reference view image for Zero123++ via a control-net [54] and use MVS-Texturing to generate a texture map. It can be observed that our approach is better in detail and global consistency than Zero123++. We compare MVS-Texturing with our optimization under the same dataset generated from stage I. Both MVS-Texturing and our approach solve an MRF problem to obtain a textured output. However, our approach yields more visually appealing results, due to view selection and the alternating optimization between view alignment and view stitching.

As shown in Tab. 1, the quantitative results are consistent with the relative improvement in visual performance. The blue numbers mean the second best performance. This result quantitatively justifies our design that combines view generation, selection, alignment, and view stitching.

Table 1: FID is computed w.r.t. the set of images synthesized by control-net. User study: numerical scores in four aspects assigned between 1 and 10 (higher is better).

Method	FID($\downarrow$)	User study Natural Color ($\uparrow$)	More Details ($\uparrow$)	Less Artifacts ($\uparrow$)	Align with Prompt ($\uparrow$)
Latent-NeRF	59.71	5.43	4.38	4.84	5.56
TEXTure	44.70	7.29	**7.92**	6.76	**8.45**
Text2Tex	39.35	7.56	7.33	6.91	7.68
SynvMVD	33.92	8.65	7.21	7.97	8.32
Zero123++	53.93	6.80	5.83	5.44	8.09
MVS-Texturing	37.47	7.36	6.46	5.51	8.34
Ours	**29.26**	**8.92**	7.49	**8.23**	8.35

Ablation Study	No view selection	No view alignment	No alternating optimization	Ours
FID($\downarrow$)	41.59	35.24	31.55	29.26

In the user study, 87 participants participated in the comparison according to four metrics, including the naturalness of the coloring ("natural color"), the details of textures ("more details"), the artifacts ("less artifacts"), and the alignment with the input prompt ("align with prompt"), with each participant undertaking 3 tests.

Our approach takes 10 to 15 minutes to texturize each model. On average, stage I takes 2 to 6 minutes, stage II takes 2.3 minutes, stage III takes 5.2 minutes, and stage IV takes 2 minutes. A desktop with a CPU of 3.2GHz and NVIDIA 4090 GPU (only for stage I) is used in our experiments. The post-processing stages (stages II-IV) take approximately twice the running time of stage I, which is comparable to baseline approaches. Increasing the efficiency of stages II to IV, possibly using neural networks, GPU-based parallelism, or a combination of the two, is subject to future research.

5.3   Ablation Study

This section describes an ablation study of our approach. Quantitative results are given in and Tab. 1 and a qualitative result is available in supp. material.
No view selection. View alignment is a critical component of our approach. Without view selection, the FID score increases by 42%. Our observation is that without view selection, the textured mesh has appearance artifacts from erroneously selected images. There are also regions where the texture appearance is discontinuous. The latter issue is caused by inconsistent image alignment, where input affects output, despite the fact that robust alignment is used.
No view alignment. View alignment is another critical component of our approach. Without view alignment, the FID score increases by 20%. In this case, there are discontinuities across the textured mesh. These effects are caused by local multiview inconsistencies among the selected views. This issue cannot be easily addressed by solving an MRF problem to identify the cuts across the most consistent regions.

No alternating optimization. Without alternating optimization, the performance of our approach drops both qualitatively and quantitatively. The FID score increases by 8%. As shown in Fig. 6(a), we see some discontinuities across the cut. These are caused by the view alignment, in which pairwise alignments take into account all overlapping regions, even though some regions disagree. With alternating optimization, view alignments tend to be more consistent along the cuts, leading to improved results after cutting.

6   Limitations

One limitation of our approach is that we solve an optimization problem to adjust the colors of the generated images to make them consistent across different views. This procedure does not understand the latent factors of the illumination changes. In the future, we plan to learn a network to adjust the color of each image conditioned on a latent parameter, where the latent parameters are optimized together to ensure view consistency among the adjusted images.

Another limitation is that pairwise image alignment and joint image alignment are decoupled, and we observed that the joint alignment stage does not perfectly fit the pairwise alignment results. This issue can be addressed by performing joint alignment to match the features of the pixels.

The third limitation is the computational cost, where stages II to IV take 10 to 15 minutes on a standard desktop. This is about two times the cost of stage I. In the future, we plan to accelerate view alignment using neural networks.

7   Conclusions

This paper introduces an optimization procedure to generate a textured 3D mesh from pre-trained text-to-image models. We study three types of multi-view inconsistency patterns among text-to-image output, which are used to develop our four-step optimization procedure. We show the design of view generation, view selection, view alignment, and view stitching. In particular, alternating between view alignment and view stitching demonstrates significant quality enhancement. The experimental results show that our approach outperforms state-of-the-art text-to-textured-mesh approaches both qualitatively and quantitatively.

There are ample venues for future research. First, unlike many existing approaches that are trained end-to-end, our approach can be considered as a postprocessing step. A natural question is whether it is possible to integrate different stages together into a single neural network to optimize its performance. Another direction is to combine pre-trained text-to-image models with training data offered by textured 3D models. Finally, controlling the textured mesh using latent parameters is another promising direction to reduce the uncertainties in text prompt specifications.

References

1. Cao, T., Kreis, K., Fidler, S., Sharp, N., Yin, K.: Texfusion: Synthesizing 3d textures with text-guided image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4169–4181 (October 2023)
2. Chen, D.Z., Siddiqui, Y., Lee, H.Y., Tulyakov, S., Nießner, M.: Text2tex: Text-driven texture synthesis via diffusion models. arXiv preprint arXiv:2303.11396 (2023)
3. 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) (October 2023)
4. Chen, Y., Chen, R., Lei, J., Zhang, Y., Jia, K.: Tango: Text-driven photorealistic and robust 3d stylization via lighting decomposition (Oct 2022)
5. Christie, M., Olivier, P., Normand, J.: Camera control in computer graphics. Comput. Graph. Forum **27**(8), 2197–2218 (2008). `https://doi.org/10.1111/j.1467-8659.2008.01181.x`, `https://doi.org/10.1111/j.1467-8659.2008.01181.x`
6. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002). `https://doi.org/10.1109/34.1000236`, `https://doi.org/10.1109/34.1000236`
7. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects (Dec 2022)
8. Deng, K., Omernick, T., Weiss, A., Ramanan, D., Zhu, J.Y., Zhou, T., Agrawala, M.: Flashtex: Fast relightable mesh texturing with lightcontrolnet (2024)
9. Dong, Y., Zuo, Q., Gu, X., Yuan, W., Zhao, Z., Dong, Z., Bo, L., Huang, Q.: Gpld3d: Latent diffusion of 3d shape generative models by enforcing geometric and physical priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 56–66 (June 2024)
10. Dutagaci, H., Cheung, C.P., Godil, A.: A benchmark for best view selection of 3d objects. In: Proceedings of the ACM Workshop on 3D Object Retrieval. p. 45–50. 3DOR '10, Association for Computing Machinery, New York, NY, USA (2010). `https://doi.org/10.1145/1877808.1877819`, `https://doi.org/10.1145/1877808.1877819`
11. Efros, A.A., Freeman, W.T.: Image Quilting for Texture Synthesis and Transfer. Association for Computing Machinery, New York, NY, USA, 1 edn. (2023), `https://doi.org/10.1145/3596711.3596771`
12. Guo, Y., Zuo, X., Dai, P., Lu, J., Wu, X., Cheng, L., Yan, Y., Xu, S., Wu, X.: Decorate3d: Text-driven high-quality texture generation for mesh decoration in the wild. In: Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS) (2023)
13. Hamdi, A., Giancola, S., Ghanem, B.: MVTN: multi-view transformation network for 3d shape recognition. In: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021. pp. 1–11. IEEE (2021). `https://doi.org/10.1109/ICCV48922.2021.00007`, `https://doi.org/10.1109/ICCV48922.2021.00007`
14. Hamdi, A., Giancola, S., Ghanem, B.: Voint cloud: Multi-view point cloud representation for 3d understanding. In: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net (2023), `https://openreview.net/pdf?id=IpGgfpMucHj`

15. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
16. Kanezaki, A., Matsushita, Y., Nishida, Y.: Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 5010–5019. Computer Vision Foundation / IEEE Computer Society (2018). `https://doi.org/10.1109/CVPR.2018.00526`, `http://openaccess.thecvf.com/content_cvpr_2018/html/Kanezaki_RotationNet_Joint_Object_CVPR_2018_paper.html`
17. Kappes, J.H., Andres, B., Hamprecht, F.A., Schnörr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B.X., Kröger, T., Lellmann, J., Komodakis, N., Savchynskyy, B., Rother, C.: A comparative study of modern inference techniques for structured discrete energy minimization problems. International Journal of Computer Vision pp. 1–30 (2015). `https://doi.org/10.1007/s11263-015-0809-x`, `http://dx.doi.org/10.1007/s11263-015-0809-x`
18. Kim, S., Tai, Y., Lee, J., Park, J., Kweon, I.S.: Category-specific salient view selection via deep convolutional neural networks. Comput. Graph. Forum **36**(8), 313–328 (2017). `https://doi.org/10.1111/cgf.13082`, `https://doi.org/10.1111/cgf.13082`
19. Knodt, J., Gao, X.: Consistent latent diffusion for mesh texturing (2023)
20. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. IEEE Trans. Pattern Anal. Mach. Intell. **28**(10), 1568–1583 (2006). `https://doi.org/10.1109/TPAMI.2006.200`, `https://doi.org/10.1109/TPAMI.2006.200`
21. Kundu, A., Yin, X., Fathi, A., Ross, D.A., Brewington, B., Funkhouser, T.A., Pantofaru, C.: Virtual multi-view fusion for 3d semantic segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIV. Lecture Notes in Computer Science, vol. 12369, pp. 518–535. Springer (2020). `https://doi.org/10.1007/978-3-030-58586-0_31`, `https://doi.org/10.1007/978-3-030-58586-0_31`
22. Lee, C.H., Varshney, A., Jacobs, D.W.: Mesh saliency. In: ACM SIGGRAPH 2005 Papers. p. 659–666. SIGGRAPH '05, Association for Computing Machinery, New York, NY, USA (2005). `https://doi.org/10.1145/1186822.1073244`, `https://doi.org/10.1145/1186822.1073244`
23. Leifman, G., Shtrom, E., Tal, A.: Surface regions of interest for viewpoint selection. IEEE Trans. Pattern Anal. Mach. Intell. **38**(12), 2544–2556 (2016). `https://doi.org/10.1109/TPAMI.2016.2522437`, `https://doi.org/10.1109/TPAMI.2016.2522437`
24. Liu, C., Yuen, J., Torralba, A.: SIFT flow: Dense correspondence across scenes and its applications. IEEE Trans. Pattern Anal. Mach. Intell. **33**(5), 978–994 (2011). `https://doi.org/10.1109/TPAMI.2010.147`, `https://doi.org/10.1109/TPAMI.2010.147`
25. Liu, R., Wu, R., Hoorick, B.V., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. CoRR **abs/2303.11328** (2023). `https://doi.org/10.48550/ARXIV.2303.11328`, `https://doi.org/10.48550/arXiv.2303.11328`
26. Liu, Y., Lin, C., Zeng, Z., Long, X., Liu, L., Komura, T., Wang, W.: Syncdreamer: Generating multiview-consistent images from a single-view image. CoRR **abs/2309.03453** (2023). `https://doi.org/10.48550/ARXIV.2309.03453`, `https://doi.org/10.48550/arXiv.2309.03453`

27. Liu, Y., Xie, M., Liu, H., Wong, T.T.: Text-guided texturing by synchronized multi-view diffusion. arXiv preprint arXiv:2311.12891 (2023)
28. Metzer, G., Richardson, E., Patashnik, O., Giryes, R., Cohen-Or, D.: Latent-nerf for shape-guided generation of 3d shapes and textures. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023. pp. 12663–12673. IEEE (2023). `https://doi.org/10.1109/CVPR52729.2023.01218`, `https://doi.org/10.1109/CVPR52729.2023.01218`
29. Michel, O., Bar-On, R., Liu, R., Benaim, S., Hanocka, R.: Text2mesh: Text-driven neural stylization for meshes. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2022). `https://doi.org/10.1109/cvpr52688.2022.01313`, `http://dx.doi.org/10.1109/cvpr52688.2022.01313`
30. Mohammad Khalid, N., Xie, T., Belilovsky, E., Popa, T.: Clip-mesh: Generating textured meshes from text using pretrained image-text models. In: SIGGRAPH Asia 2022 Conference Papers (Nov 2022). `https://doi.org/10.1145/3550469.3555392`, `http://dx.doi.org/10.1145/3550469.3555392`
31. 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. OpenReview.net (2023), `https://openreview.net/pdf?id=FjNys5c7VyY`
32. Qiu, L., Chen, G., Gu, X., zuo, Q., Xu, M., Wu, Y., Yuan, W., Dong, Z., Bo, L., Han, X.: Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. arXiv preprint arXiv:2311.16918 (2023)
33. Richardson, E., Metzer, G., Alaluf, Y., Giryes, R., Cohen-Or, D.: Texture: Text-guided texturing of 3d shapes. In: ACM SIGGRAPH 2023 Conference Proceedings. SIGGRAPH '23, Association for Computing Machinery, New York, NY, USA (2023). `https://doi.org/10.1145/3588432.3591503`, `https://doi.org/10.1145/3588432.3591503`
34. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2022). `https://doi.org/10.1109/cvpr52688.2022.01042`, `http://dx.doi.org/10.1109/cvpr52688.2022.01042`
35. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S., Crowson, K., Schmidt, L., Kaczmarczyk, R., Jitsev, J.: LAION-5B: an open large-scale dataset for training next generation image-text models. In: NeurIPS (2022), `http://papers.nips.cc/paper_files/paper/2022/hash/a1859debfb3b59d094f3504d5ebb6c25-Abstract-Datasets_and_Benchmarks.html`
36. Secord, A., Lu, J., Finkelstein, A., Singh, M., Nealen, A.: Perceptual models of viewpoint preference. ACM Trans. Graph. **30**(5) (oct 2011). `https://doi.org/10.1145/2019627.2019628`, `https://doi.org/10.1145/2019627.2019628`
37. Sederberg, T.W., Parry, S.R.: Free-form deformation of solid geometric models. SIGGRAPH Comput. Graph. **20**(4), 151–160 (aug 1986). `https://doi.org/10.1145/15886.15903`, `https://doi.org/10.1145/15886.15903`
38. Shi, R., Chen, H., Zhang, Z., Liu, M., Xu, C., Wei, X., Chen, L., Zeng, C., Su, H., San, U., University, S.: Zero123++: a single image to consistent multi-view diffusion base model
39. 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)

40. Soltani, A.A., Huang, H., Wu, J., Kulkarni, T.D., Tenenbaum, J.B.: Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 2511–2519. IEEE Computer Society (2017). `https://doi.org/10.1109/CVPR.2017.269`, `https://doi.org/10.1109/CVPR.2017.269`

41. Song, R., Zhang, W., Zhao, Y., Liu, Y.: Unsupervised multi-view CNN for salient view selection and 3d interest point detection. Int. J. Comput. Vis. **130**(5), 1210–1227 (2022). `https://doi.org/10.1007/s11263-022-01592-x`, `https://doi.org/10.1007/s11263-022-01592-x`

42. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.G.: Multi-view convolutional neural networks for 3d shape recognition. In: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015. pp. 945–953. IEEE Computer Society (2015). `https://doi.org/10.1109/ICCV.2015.114`, `https://doi.org/10.1109/ICCV.2015.114`

43. Sun, Y., Huang, Q., Hsiao, D., Guan, L., Hua, G.: Learning view selection for 3d scenes. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. pp. 14464–14473. Computer Vision Foundation / IEEE (2021). `https://doi.org/10.1109/CVPR46437.2021.01423`, `https://openaccess.thecvf.com/content/CVPR2021/html/Sun_Learning_View_Selection_for_3D_Scenes_CVPR_2021_paper.html`

44. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Multi-view 3d models from single images with a convolutional network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VII. Lecture Notes in Computer Science, vol. 9911, pp. 322–337. Springer (2016). `https://doi.org/10.1007/978-3-319-46478-7_20`, `https://doi.org/10.1007/978-3-319-46478-7_20`

45. Tsalicoglou, C., Manhardt, F., Tonioni, A., Niemeyer, M., Tombari, F.: Textmesh: Generation of realistic 3d meshes from text prompts (2023)

46. Waechter, M., Moehrle, N., Goesele, M.: Let there be color! large-scale texturing of 3d reconstructions. In: Fleet, D.J., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V. Lecture Notes in Computer Science, vol. 8693, pp. 836–850. Springer (2014). `https://doi.org/10.1007/978-3-319-10602-1_54`, `https://doi.org/10.1007/978-3-319-10602-1_54`

47. Wei, X., Yu, R., Sun, J.: View-gcn: View-based graph convolutional network for 3d shape analysis. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 1847–1856. Computer Vision Foundation / IEEE (2020). `https://doi.org/10.1109/CVPR42600.2020.00192`, `https://openaccess.thecvf.com/content_CVPR_2020/html/Wei_View-GCN_View-Based_Graph_Convolutional_Network_for_3D_Shape_Analysis_CVPR_2020_paper.html`

48. Weng, H., Yang, T., Wang, J., Li, Y., Zhang, T., Chen, C.L.P., Zhang, L.: Consistent123: Improve consistency for one image to 3d object synthesis. CoRR **abs/2310.08092** (2023). `https://doi.org/10.48550/ARXIV.2310.08092`, `https://doi.org/10.48550/arXiv.2310.08092`

49. Xu, Y., Tan, H., Luan, F., Bi, S., Wang, P., Li, J., Shi, Z., Sunkavalli, K., Wetzstein, G., Xu, Z., Zhang, K.: Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model (2023)

50. Ye, J., Wang, P., Li, K., Shi, Y., Wang, H.: Consistent-1-to-3: Consistent image to 3d view synthesis via geometry-aware diffusion models (2023)
51. Youwang, K., Oh, T.H., Pons-Moll, G.: Paint-it: Text-to-texture synthesis via deep convolutional texture map optimization and physically-based rendering. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2024)
52. Yu, X., Dai, P., Li, W., Ma, L., Liu, Z., Qi, X.: Texture generation on 3d meshes with point-uv diffusion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4206–4216 (October 2023)
53. Zeng, X., Chen, X., Qi, Z., Liu, W., Zhao, Z., Wang, Z., FU, B., Liu, Y., Yu, G.: Paint3d: Paint anything 3d with lighting-less texture diffusion models (2023)
54. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3836–3847 (October 2023)
55. Zuo, Q., Gu, X., Qiu, L., Dong, Y., Zhao, Z., Yuan, W., Peng, R., Zhu, S., Dong, Z., Bo, L., Huang, Q.: Videomv: Consistent multi-view generation based on large video generative model (2024)