GenVDM: Generating Vector Displacement Maps From a Single Image

Anonymous CVPR submission

Paper ID 10476

Abstract

We introduce the first method for generating Vector Displacement Maps (VDMs): parameterized, detailed geometric stamps commonly used in 3D modeling. Given a single input image, our method first generates multi-view normal maps and then reconstructs a VDM from the normals via a novel reconstruction pipeline. We also propose an efficient algorithm for extracting VDMs from 3D objects, and present the first academic VDM dataset. Compared to existing 3D generative models focusing on complete shapes, we focus on generating parts that can be seamlessly attached to shape surfaces. The method gives artists rich control over adding geometric details to a 3D shape. Experiments demonstrate that our approach outperforms existing baselines. Generating VDMs offers additional benefits, such as using 2D image editing to customize and refine 3D details.

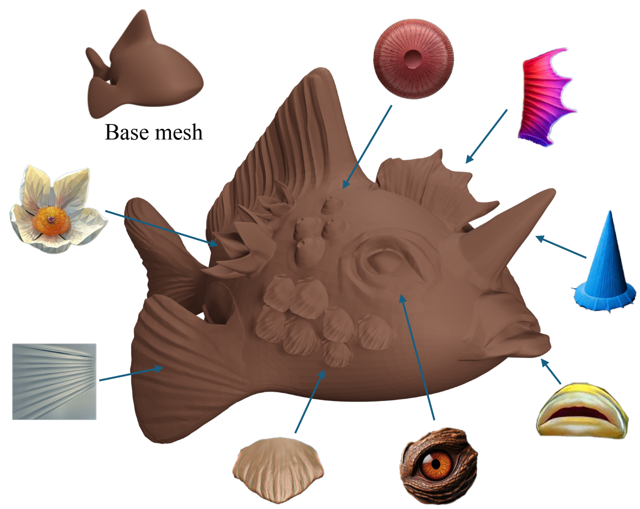


Figure 1. We introduce GenVDM, a method that can generate a highly detailed Vector Displacement Map (VDM) from a single input image. The generated VDMs can be directly applied to mesh surfaces to create intricate geometric details. Note that the thumbnails represent plain 2D RGB image sources.

1. Introduction

Generative neural models for 3D shape synthesis is a rapidly advancing research area [58]. However, they are still not widely adopted in artistic workflows for two main reasons. First, synthesizing fine geometric details is challenging due to the heterogeneity of 3D representations and the lack of detailed 3D training data. Second, existing neural tools lack the precise spatial and compositional controls needed by 3D artists. To address these limitations, instead of reinventing the 3D modeling stack to accommodate generative AI, we draw inspiration from an existing workflow in which an artist starts with a base mesh and "stamps" the desired details onto the 3D surface (see Figure 1). These smaller stamps are easier to generate than full-scale 3D models, fit seamlessly into existing workflows, eliminate artists' dependence on expensive and limited third-party stamp libraries, and provide full artistic control over spatial arrangement and composition.

We chose the *vector displacement map* or VDM as our stamp representation. A VDM assigns an arbitrary 3D displacement to every point in a 2D rectangle, warping the sheet to form a curved surface with complex geometric features, such as overhangs and cavities. It is widely supported in 3D software [1–4] and compactly stored as a vector field over a UV image domain. While using VDMs is commonplace, authoring them is extremely challenging, and artists usually depend on packs of VDMs created by third parties (analogous to brushes in digital painting tools), with limited customization or generality. Image or text-driven stamp generation could drastically expand the scope of VDM usage by providing artists with custom stamps on demand.

In this paper, we propose the first neural pipeline to generate a VDM from a single RGB image. To achieve this, we address two main technical challenges. The first challenge is that existing generative models are not suitable for VDM generation: generating a 3D object usually does not also produce a parametric 2D domain for stamp application, and predicting a depth map from a single image does not capture complex high-amplitude variations, overhangs, and occlusions; see Figure 6. Thus, we develop a three-step method. First, given an input RGB image (which can also be generated with existing text-to-image models), we predict normal maps from multiple viewing directions to resolve occlusions that may be hidden in a single view. Second, we reconstruct a mesh (which need not have disk topology) by fitting a neural SDF to the multi-view normal maps and polygonizing the result. Third, we use a neural deformation model to displace points on a 2D rectangle to fit the mesh, forming the final VDM.

The second challenge in training a generative VDM model is the absence of training data. We tackle it by building an interactive tool to segment interesting semantic and geometric regions from Objaverse 3D models [19], and then develop a geometry processing pipeline for converting these regions into a VDM representation, creating a dataset of 1,200 VDM patches used for training. Our pipeline is robust enough to analyze polygon soups in the wild, which we achieve by re-sampling the selected regions and reconstructing a single connected surface after removing outliers. We then deform the resulting mesh to obtain a co-planar boundary that can be seamlessly attached to a flat base tile over which the VDMs are typically defined. The processed shapes can then be rendered and used to finetune the multi-view normal generation model.

We compare our method to state-of-the-art shape generation techniques [27, 40, 51], as well as to reconstructing a heightfield (i.e. a *scalar* displacement map) from estimated depth [81]. We use a collection of images depicting parts commonly used in VDMs (e.g., facial elements, decorations), and evaluate using visual fidelity [54] and semantic similarity [52] metrics. Our method outperforms others due to its ability to handle smaller VDM-like regions. Note also that other mesh generation methods do not produce a displacement map – which can have both "outward" and "inward" displacements – and thus their output can only be additively combined with the base shape, e.g., they are not able to introduce cavities like an eye or a mouth in Figure 1.

To summarize, our contributions are:

- The first generative ML pipeline for VDMs;
- A robust method to reconstruct VDMs from multi-view normal maps produced by image diffusion models;
- A novel VDM extraction pipeline to efficiently extract and process patches from 3D objects to produce VDMs;
- The first public dataset of VDMs for academic research.

2. Related work

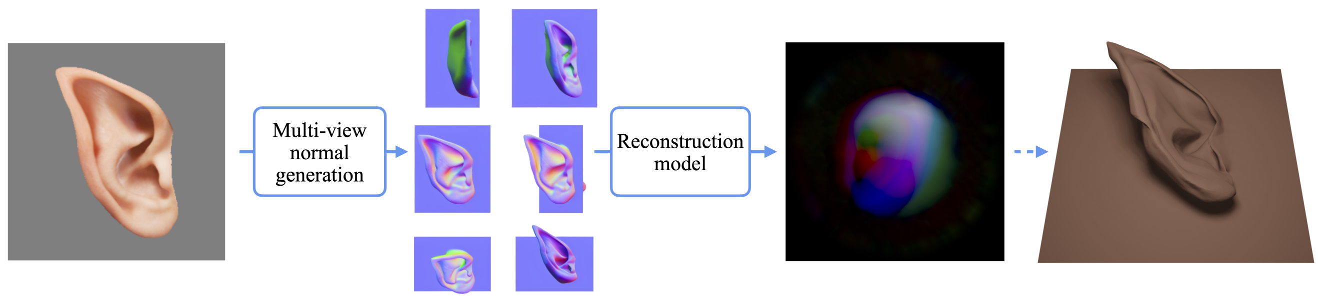
Vector Displacement Maps. Texture mapping [10, 26] is the dominant solution in the industry to add complex surface details to shapes without increasing mesh complexity. Accompanying it are many techniques that hallucinate complex geometric details, such as bump mapping [9], horizon mapping [43], and parallax mapping [30]. Unlike those techniques that do not change the geometry of the shape, displacement mapping [17, 18, 61] adds geometric details by subdividing the original geometry into finer polygons and then displacing each vertex in its normal direction by a height value indexed from the displacement map (although some versions of displacement mapping can be done in the pixel space without changing the original geometry [66]).

While a displacement map can be considered as a single-channel image or heightfield, a vector displacement map (VDM) can be seen as a three-channel image, where each pixel contains a 3D displacement vector. VDMs naturally support representing more complex geometries with less distortion compared to displacement maps, and both are used in 3D modeling tools to create geometric details. Research on displacement maps and VDMs has focused on texture synthesis from examples [82], and synthesis of human body and face meshes for shape reconstruction [6, 80]. VDMs conceptually resemble Geometry Images [23], and some recent works adopt image diffusion models for generating Geometry Images to synthesize 3D shapes [20, 79]. To our knowledge, there is no prior work on generative models of VDMs, nor a public research dataset for VDMs.

Image-to-3D. Early works on single-view 3D reconstruction [15, 16, 22, 45, 67, 78, 83] mostly adopt feed-forward neural networks trained on limited data [11]. More recent work [29, 46, 85, 87] trained on large 3D datasets [19] has shown significantly improved generalizability to novel shape categories. With the introduction of text-to-image diffusion models [49, 53], a line of work [44, 63] achieved zero-shot single-image-to-3D with score distillation sampling (SDS) [50] by distilling 2D diffusion priors into 3D representations with per-shape optimization.

Another line of work [38, 71] utilizes image diffusion models for novel view synthesis conditioned on an input image and a relative camera pose. Such models produce images of the object from different views, therefore the 3D object can be reconstructed by SDS-based optimization [38, 51] or a feed-forward reconstruction network [37]. These methods inspired a series of subsequent work that finetunes pretrained image diffusion models to directly generate 3D-consistent multi-view images of the target output shape given a single-view image, where the output shape can be reconstructed from generated multi-view images via optimizing a neural field or mesh [39, 40, 57], a 3D diffusion reconstruction network [36], or a feed-forward large reconstruction model powered by Transformers [27, 34, 64, 68, 70, 72, 74, 76, 86, 88]. Most recently, image diffusion models have been replaced by video diffusion models to achieve better 3D consistency of the generated views [24, 65].

Modeling by Parts. The use of small building components to compose complex shapes has been widely studied in modeling-by-assembly systems [21, 32]. Before gener-



(a) Input RGB image (b) Multi-view normal images (c) VDM image (2D) (d) VDM applied to a plane mesh (3D)

Figure 2. Overview of our image-to-VDM pipeline. Given an input image, we first add a gray square behind the object/part in the image as background, so the image resembles a textured VDM applied to a square mesh, as in (a). Then we utilize a multi-view image diffusion model to generate six normal maps with pre-defined camera poses, as in (b). The multi-view normal maps effectively represent the geometry of the VDM when applied to a square mesh, and thus we can reconstruct the VDM from these normal maps, as in (c). The reconstructed VDM can then be applied to various surfaces as in (d).

ative AI rose to prominence, these systems relied on part databases [12] (or shape databases from which parts could be cut out), and focused on building tools to help users find the right parts [7, 13, 56, 75] and assemble them meaningfully [28, 60, 77]. As a variation, methods were developed to extract and transfer detailed patches from a shape to another [62]. A few papers studied joint synthesis and layout of parts [35], but the synthesis was conditioned only on the layout and not on user input, and the focus was on whole-shape generation and not adding detail to existing ones.

Relying on existing part datasets or part generation without user control, and on complex, non-standard, topology-sensitive mesh fusion algorithms limits the utility of these older methods. Our approach generates detailed complementary geometry in-situ from the image prompt, and our generated VDMs are defined over parameterized 2D domains which are suitable for seamlessly blending onto 3D models, with industry-wide support.

3. Method

Our image-to-VDM pipeline is shown in Figure 2. Similar to other methods in the literature, we follow an approach that first generates multi-view images of the target object with an image diffusion model and then reconstructs the object from the generated images. In particular, we only generate normal maps of the object as we are only interested in the geometric details. Details of the multi-view normal generation are described in Section 3.1. Next, we reconstruct the VDM from the multi-view normals. As VDMs have specific properties and constraints, reconstructing them is highly non-trivial. We report our attempts and solutions in Section 3.2. Finally, as there is no publicly available dataset for VDMs, we designed an efficient tool for extracting shape patches from Objaverse [19], and devised algorithms to process those patches for use as training data. We describe the data processing pipeline in Section 3.3.

3.1. Multi-View Normal Map Generation

We opt to finetune an image diffusion model to generate multi-view images, as the pretrained image diffusion model offers strong generalizability. As will be shown in our experiments, our model, trained on a small dataset of 1,200 examples, works on a large variety of shapes.

Specifically, we adopt Zero123++ [57] as the backbone for our multi-view diffusion model. Zero123++ is an image-to-multiview model based on Stable Diffusion [53]. Given an input image, Zero123++ generates a 960×640 image representing six multi-view images in a 3×2 grid, where the six images have pre-defined camera poses so they can be easily used for 3D reconstruction. However, the pre-defined camera poses in Zero123++ fully surround the object, e.g., there are front views and back views of the object. In our pipeline, since we are aiming to generate VDMs, the back views of the object are unnecessary. Therefore, we re-designed the camera poses of the six images. As shown in Figure 2 (b), assuming the front view (see (a) for an example) has (elevation angle, azimuth angle) $= (0°, 0°)$, we define the six camera poses to be $(0°, -60°)$, $(0°, -30°)$, $(0°, 30°)$, $(0°, 60°)$, $(45°, 0°)$, $(-45°, 0°)$. We also adopt orthographic cameras to reduce distortion, and let the model generate a normal map of the object for each camera pose. To train the model, we render single-view RGB images as input and multi-view normal maps as ground truth output. Details about training data is described in Section 3.3. Note that the input image does not have to be a front view; we render random views for training so the model can handle images from various viewpoints. We finetune on the checkpoint provided by Zero123++ [57] on 8 NVIDIA A100 GPUs for 3 days.

3.2. VDM Reconstruction

Reconstructing 3D shapes from multi-view images has been well studies in the text/image-to-3D literature. Most recent

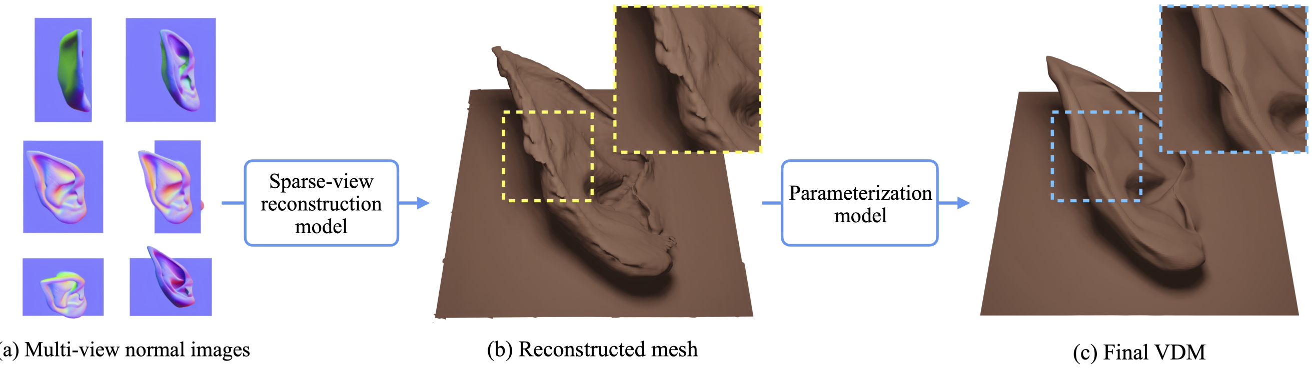


Figure 3. Reconstructing VDM from multi-view normal maps. We adopt a two-step approach. First, we reconstruct an accurate (but perhaps noisy) mesh (b) from the multi-view normals (a) with differentiable rendering and neural SDF representation. Then we parameterize the mesh by fitting a deformable square to it with a neural deformation field, as in (c). An VDM image can thus be obtained by discretizing the square into pixels and infer each pixel's displacement from the neural deformation field. The whole reconstruction pipeline takes about 6 minutes for each shape on an NVIDIA A100 GPU, where each step takes about 3 minutes.

methods adopt a feed-forward large reconstruction model (LRM) to directly generate a 3D shape from multiple input images of different viewpoints [27, 34, 64, 68, 72, 86]. Therefore, a straightforward way for reconstructing VDMs is to train a similar LRM to take the normal maps as input and directly regress a VDM image. However, given limited VDM training shapes, our LRM trained on a small dataset is unlikely to generalize as well as other LRM models trained on larger datasets, therefore leading to suboptimal results.

Given the discussions above, we adopt a slower but more robust per-shape optimization approach. Given the six normal maps with pre-defined fixed camera poses, we want to optimize a 3D representation to converge to the target 3D shape with supervision provided by differentiable rendering. A naive approach would be to initialize with a discretized square mesh and optimize with mesh-based differentiable rendering. However, as has been shown in other methods [33, 47], differentiable rendering on meshes is often problematic and requires careful design of regularization losses and tuning of hyperparameters. As we will show later, even with ground truth 3D supervision, optimizing a discretized mesh to fit the target shape is not an easy task.

Therefore, we devise a two-step approach, as shown in Figure 3, to first optimize a neural SDF field to reconstruct a 3D shape from the multi-view normal maps, and then parameterize the 3D shape into a VDM image. We utilize the method proposed in Wonder3D [40] for the first step, with the only modification being that we removed L_{rgb}, the loss term to punish the difference between rendered RGB images and the ground truth, as we do not predict multi-view RGB images. Since we always put a grey square as background in our input images, the shape we obtained via optimization has a solid plane-like primitive where the object/part is attached to, see Figure 3 (b); then we can extract a mesh from the neural SDF field and easily separate a single layer of mesh that represents the VDM.

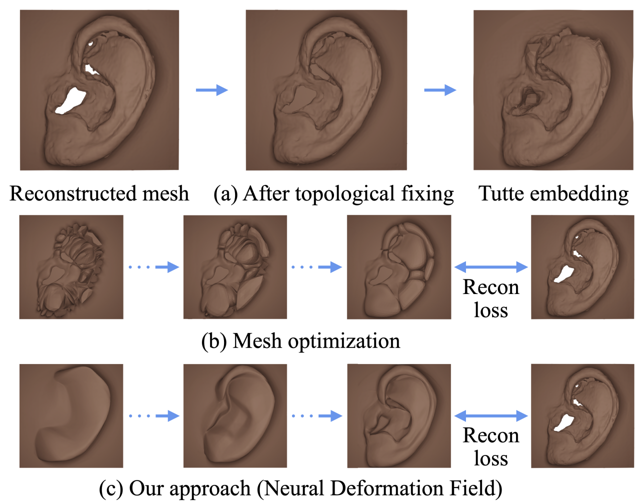
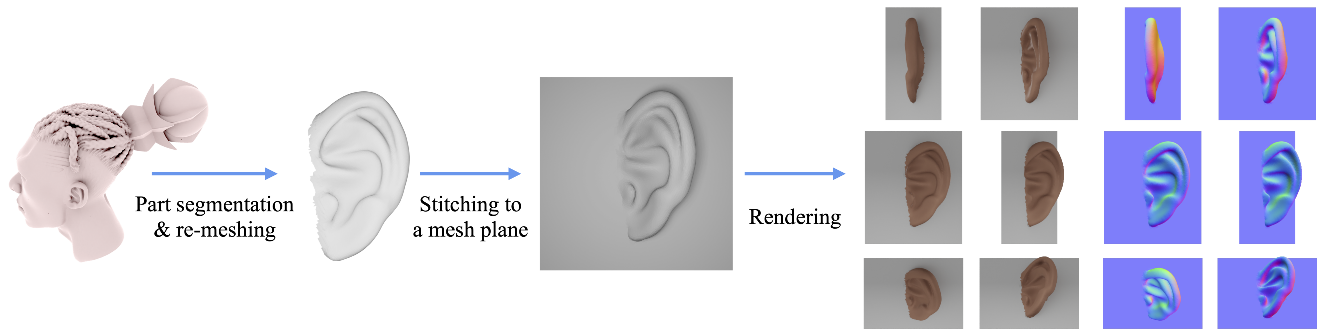


Figure 4. Comparison of different approaches for parameterizing a shape into VDM. (a) Topology fixing and Tutte embedding with classic tools leads to noise and distortion. (b) Fitting a plane mesh to the target mesh leads to large distortion. (c) Our approach by applying a neural deformation field to a parametric square leads to clean and high-quality reconstruction.

The next step is to parameterize the mesh into a VDM image. Since the mesh is reconstructed from sparse-view images, its geometry is often noisy and riddled with small holes and large gaps, see Figure 4 (a) left. To convert it into a VDM, we will need to fix its topology so that it is topologically equivalent to a plane; and then we will apply a mesh parametrization method to obtain its Tutte embedding on a square, so that each pixel on the square can be assigned with a displacement vector. However, as shown in Figure 4 (a), although the state-of-the-art topology fixing algorithms [84] can fix the topology, the result is often not satisfactory, e.g., a gap that should have been filled is being cut, see Figure 4 (a) middle where the helix of the ear is cut in half. As a result, after applying [55] to obtain its



(a) Object mesh (b) Part mesh (c) Part stitched to a square mesh (d) RGB images (e) Normal maps

Figure 5. Data preparation. For each interesting object (a), we use a 3D lasso tool to segment out interesting parts. For each part, we densely sample points on the part's surface and then perform Screened Poisson Surface Reconstruction [31] to obtain a single connected mesh (b). We then stitch the mesh to a square mesh with an algorithm inspired by Poisson Image Editing [48] (c). Afterwards, we can color the part and render RGB images (d) and normal maps (e) for training the image diffusion model.

embedding on a plane, we see large distortions and noise in the final VDM, see Figure 4 (a) right where the upper part of the ear is missing due to distortion.

An alternative is to initialize with an optimizable square mesh, and optimize it using a reconstruction loss with respect to the target mesh, as shown in Figure 4 (b). However, as mentioned, it is often required to have carefully designed regularization losses when a mesh is to be optimized. When adopting a naive optimization method proposed in [14], the resulting mesh exhibits large distortion.

Therefore, instead of tuning the mesh optimization algorithm, inspired by AtlasNet [22] and Deep Geometric Prior [73], we propose to deform the square mesh with a neural deformation field parameterized by a Multilayer Perceptron (MLP). The MLP acts as a natural regularizer, as its inductive smoothness bias encourages smoothness of the deformation. We define the square to be $\{p \mid p \in [0,1]^2\}$, and the MLP ϕ_θ with optimizable parameters θ. Then, given any 2D point p in the square, we obtain its corresponding 3D point $p' = \phi_\theta(p)$ in the deformed shape. Therefore, for each optimization step, we sample a grid of 2D points in $[0,1]^2$, apply ϕ_θ to obtain the deformed 3D points, and then compute the symmetric Chamfer Distance between the deformed 3D points and the ground truth points sampled from the target mesh. We also include a loss to maintain square boundary. Therefore our optimization objective is

$$\begin{aligned}\operatorname*{argmin}_{\theta} \quad \mathbb{E}_{P,Q} \ & \frac{1}{|P|}\sum_{p\in P}\min_{q\in Q}\|\phi_\theta(p)-q\|_2^2 + \\ & \frac{1}{|Q|}\sum_{q\in Q}\min_{p\in P}\|\phi_\theta(p)-q\|_2^2 + \\ & \frac{1}{|\partial P|}\sum_{p\in\partial P}\|\phi_\theta(p)-\text{proj}(p)\|_2^2,\end{aligned} \tag{1}$$

where P and Q are sets of sampled points from $[0,1]^2$ and the target mesh, respectively. ∂P contains all the boundary points in P and $\text{proj}(p)$ maps p to a corresponding 3D point in a pre-defined square boundary. After optimization, we can sample a regular grid of points in $[0,1]^2$ and compute their 3D displacement vectors from ϕ_θ to obtain the VDM image, as shown in Figure 4 (c).

3.3. Data Preparation

To the best of our knowledge, there is no publicly available dataset for VDMs. Therefore, we developed a data processing pipeline so we can efficiently annotate interesting parts from objects and then convert the parts into VDMs. In fact, our data processing pipeline does not produce true VDMs, but rather, shapes that look like VDMs, which are good enough for training our multi-view generation model, see Figure 5. If needed, our VDM reconstruction method in Section 3.2 can be used to obtain readily usable VDMs.

To construct our VDM training dataset, we crop parts from the Objaverse [19] dataset. We first create a keyword filtering list and apply the filter on Objaverse shape captions [41, 42]. As VDMs are mostly used to model organic parts, we select objects likely to contain such parts, e.g., animals and characters.

We then developed a UI to precisely crop a part from a 3D object. This is achieved by a 3D lasso tool, where the user only needs to select a ring of points along the cutting boundary of the desired part. Our algorithm connects the points to form a cut and extracts the part from the object. Note that the part may not be a single connected mesh – it may comprise several sub-meshes. Hence, we remesh the part into a single connected mesh. We first densely sample points on the part, and then remove interior points by computing winding numbers [8]. For the remaining points, we perform Screened Poisson Surface Reconstruction [31] to obtain a single connected mesh (Figure 5 (b)). Our 3D lasso tool has proven to be quite efficient. Annotating our entire dataset with 1,200 parts took only 24 man-hours.

After obtaining the parts, we will then stitch each part to a square mesh to mimic the appearance of a VDM applied to a plane. Note that in almost all cases, the vertices on the boundary of each part are not coplanar, therefore additional steps are required to make them coplanar. We first determine the plane via least squares plane fitting with respect to the boundary vertices. Then we project the boundary vertices to the plane, and adopt a method similar to Poisson Image Editing [48] to deform the part so that it follows the new coplanar boundary. Denote the set of all boundary vertices in the part (before projection) as B and non-boundary vertices as A; also denote the set of all edges as E. Denote the coplanar boundary vertices after projection as B', and the non-boundary vertices after deformation as A'. For each point p in A or B, denote its corresponding point in A' or B' as p'. Then our new vertices after mesh deformation can be obtained by solving a quadratic error function

$$\underset{A'}{\operatorname{argmin}} \ \mathbb{E}_{(p,q)\in E} \ \|(p'-q')-(p-q)\|_2^2. \tag{2}$$

The minimization objective is to ensure that the gradients on the mesh are preserved as much as possible after deformation, while the target coplanar boundary points B' are also strictly followed.

We then place the deformed part on a square mesh so that the boundary vertices and the square mesh vertices are coplanar. Once the part is attached to the square mesh, we perform one additional Laplacian Smoothing step to the vertices close to the boundary to remove boundary noise, see Figure 5 (c). We always keep the square mesh gray and assign a random color to the part. We also performs translation, scaling, and rotation augmentation to the part to enrich the diversity of the dataset. Finally, for each shape, we render several RGB images from different viewpoints to serve as the training input to the multi-view normal generation model, and six normal maps in pre-defined camera poses as the ground truth output, see Figure 5 (d, e).

4. Experiments

In this section, we verify the effectiveness of our method by comparing it with various state-of-the-art methods. We also validate our design choices in ablation studies. Finally, we present additional results produced by our method, show applications of VDMs on adding details to geometry, and demonstrate how users can customize VDMs by simply editing the input images. We will make our code, trained model weights, and dataset available to the public.

4.1. Vector Displacement Map Generation

Baselines. Since there is no prior work on generating VDMs from single view images, we compare our method with methods that perform a similar task, namely, single-view image to 3D reconstruction. Specifically, we compare our method with Wonder3D [40], Magic123 [51], Large Reconstruction Model (LRM) [27], as well as a *scalar* displacement map (scalar DM) reconstruction method based on DepthAnything [81]. Given an input image, Wonder3D [40] generates multi-view RGB and normal images and optimizes a neural SDF field to reconstruct the 3D shape from the multi-view images. Magic123 [51] leverages SDS loss [50] to optimize the 3D shape while applying a reconstruction loss on the input view. LRM [27] generates multi-view RGB images and trains a Transformer-based feed-forward model to reconstruct the 3D shape from the multi-view images. To validate the necessity of generating *vector* displacement map instead of regular *scalar* displacement map, we also compare with a state-of-the-art depth prediction method, DepthAnything [81], by converting the predicted depth of the object into a *scalar* DM. We run these baseline models with official implementation and pretrained weights; except that LRM does not release the official code, so we use open-source implementation OpenLRM [25] instead. For all the reconstructed shapes, we render textureless images for visualization and evaluation. For Wonder3D, Magic123, and LRM, as they generate complete objects and not VDMs, we put a square plane behind their generated shapes to make the visualization more consistent and to have a fair quantitative comparison.

Evaluation Dataset and Metrics. As there is no existing benchmark dataset for VDMs, we collected a dataset of 50 RGB images from the Internet and a text-to-image model [5] for evaluation. All images depict common VDM categories used by artists such as facial elements and decorations. For quantitative evaluation, we measure CLIP similarity [52] and 3D-FID score [69] between the input image and the rendered images of the generated shapes from different views, denoted as **CLIPImg** and **3D-FID**, respectively. For CLIP, we additionally evaluate semantic alignment by measuring CLIP similarity between the rendered images and the texts describing the categories of the input images, denoted as **CLIPText**. We use public implementation of CLIP [59] and 3D-FID [54] for computing the metrics. Please see Supplementary Material for more details.

The quantitative results are summarized in Table 1 and qualitative results are presented in Figure 6. Quantitatively, our method outperforms others by a significant margin. The closest competitors to our method are Wonder3D and scalar DM, which is also reflected in the qualitative results in Figure 6. Magic123 and LRM lack geometric detail as they rely heavily on textures which often hallucinate details in geometry. Wonder3D has a similar shape generation pipeline with ours, yet it was designed to generate complete objects. Therefore, it struggles to generate partial shapes, e.g., noses and ears. Although the results of scalar DM look reasonable from the front view, its side view suffers as scalar DM cannot represent unseen regions of the front view.

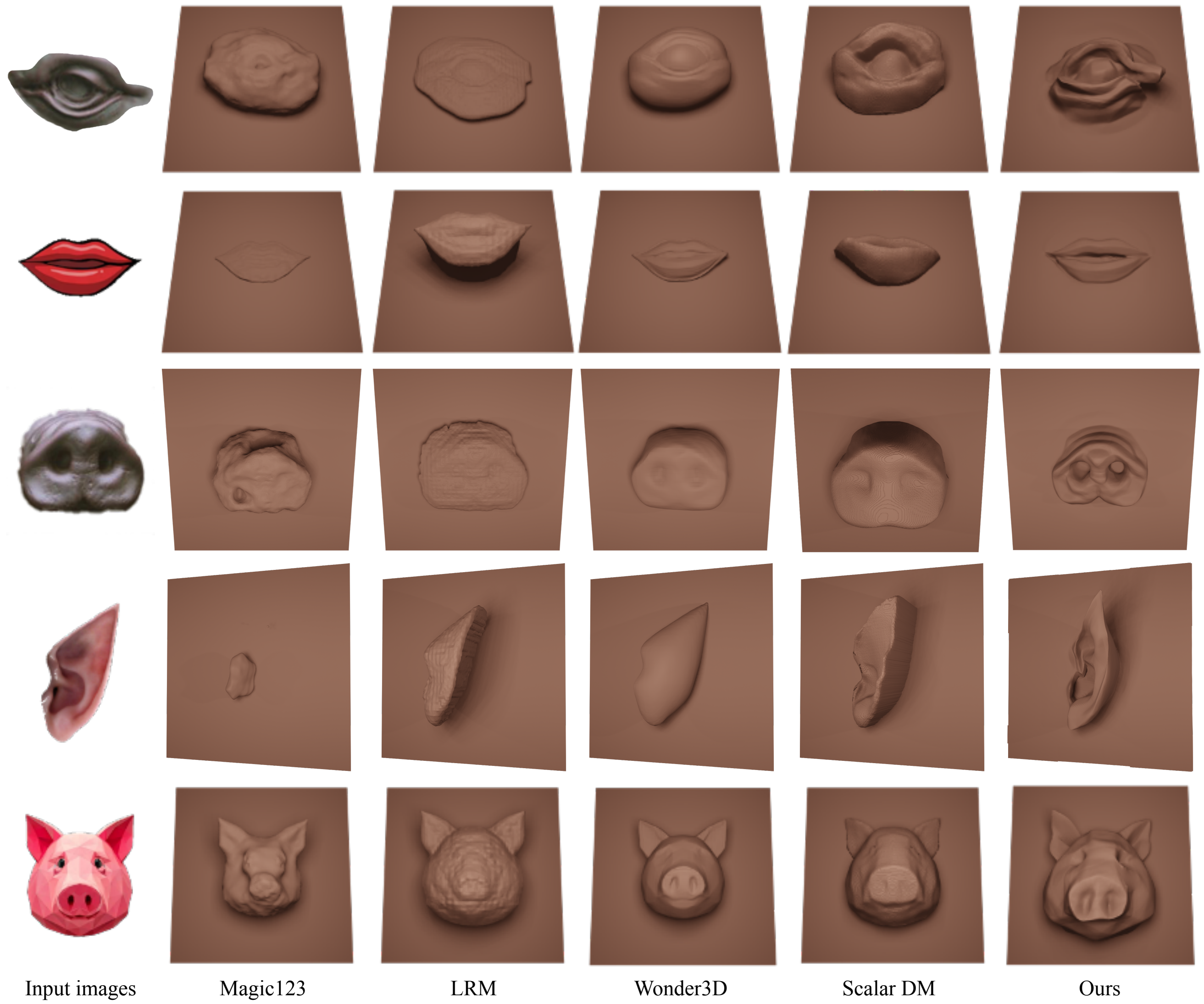


Figure 6. Qualitative results compared with baseline methods. As Magic123 [51], LRM [27], and Wonder3D [40] generate complete objects and not VDMs, we put a square plane behind their generated shapes to make the visualization more consistent.

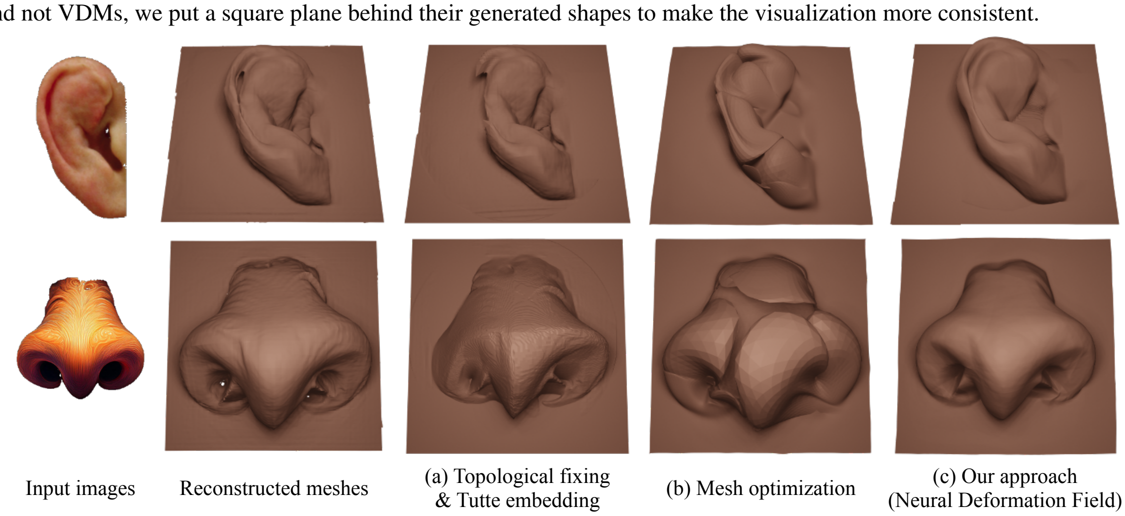


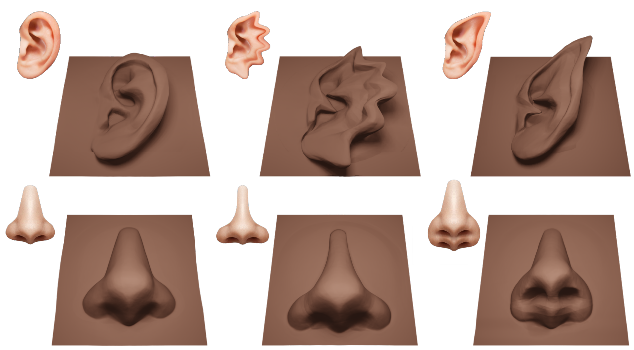
Figure 7. Qualitative results of ablation study.

Method	CLIPImg↑	CLIPText↑	3D-FID↓
Wonder3D [40]	0.8246	0.2542	199.5
Magic123 [51]	0.8293	0.2510	213.2
LRM [27]	0.8144	0.2510	239.9
Scalar DM	0.8223	0.2564	213.0
Ours	**0.8520**	**0.2701**	**192.7**

Table 1. Quantitative comparison with baseline methods. Scalar DM stands for scalar displacement map produced from DepthAnything [81].

Method	CLIPImg↑	CLIPText↑	3D-FID↓
Recon. Mesh	0.8440	0.2636	198.0
Topo. Fix(a)	0.8401	0.2617	209.9
Mesh Opt.(b)	0.8245	0.2525	217.2
Ours(c)	**0.8521**	**0.2701**	**192.7**

Table 2. Quantitative ablation on VDM Reconstruction.

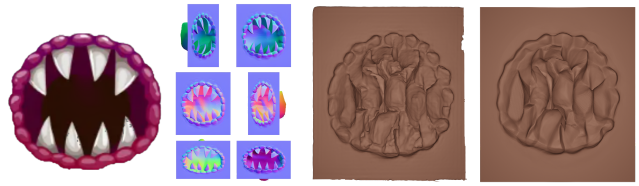


(a) Original (b) Edition 1 (c) Edition 2

Figure 8. Customizing VDMs by editing images. Here we show original input images and generated VDMs in (a) and edited images and their generated VDMs in (b)(c).

4.2. Ablation Study

As discussed in Section 3.2, we compare the following settings for parameterizing the reconstructed mesh into a VDM image: (a) Topology fixing and Tutte embedding, (b) fitting a square mesh into reconstructed mesh, and (c) our approach; see Figure 4. We also include the reconstructed mesh before parameterization as a reference baseline. Table 2 summarizes quantitative results and Figure 6 shows qualitative comparisons. Topological fixing and Tutte embedding suffer when the topology of the reconstructed mesh is complex due to noisy reconstruction results, as shown in Figure 6 (a). This is because the topological fixing algorithm does not consider the distortion after parameterization as one of its optimization goals, thus some topological fixes may significantly increase distortion. Figure 6 (b) shows that mesh optimization is not reliable in our setting, and is likely to fall into local minima during optimization. In contrast, our method, shown in Figure 6 (c), not only reconstructs high quality VDMs with correct topology, but also smooths out noise induced in neural SDF reconstruction, leading to visually more pleasing results.



(a) Input image (b) Normal maps (c) Reconstructed mesh (d) Final VDM

Figure 9. Failure case.

4.3. Application

Shape modeling. With our method, users are able to generate parts of the shape from single-view images or text prompts (via text-to-image to obtain the input to our method). Compared with methods that generate complete shapes, our method naturally provides more controllability, as users can start with a coarse shape and add customization details and shape parts, see Figure 1. We also show a video in the Supplementary Material to demonstrate the modeling process with VDMs generated by our method.

Part editing. With our image-to-VDM, one can perform editing in 2D image space and change the appearance of the part in 3D, see Figure 8. Editing in image space is typically much more convenient than sculpting 3D shapes, therefore allowing users to customize their parts with ease.

5. Conclusion, Limitation, and Future Work

In this work, we propose a method to generate a VDM from an input single-view image. Our method first finetunes a pretrained image diffusion model to generate multi-view normal maps from the input image, and then reconstructs a VDM image from the multi-view normals. The generated VDMs can be used directly in shape modeling, which provide more freedom to the users on the appearance and position of each part on the shape. We also propose an efficient pipeline for creating a VDM dataset from 3D objects. Our method outperforms state-of-the-art image-to-3D models and scalar displacement map baseline, proving that our approach is more suited for VDM generation.

As discussed in Section 3.2, our VDM reconstruction involves per-shape optimization, making its inference time significantly slower than the current image-to-3D methods with feed-forward LRM. Investigating the possibility of a VDM-LRM with limited training data is of great interest to us. For certain shapes with thin structures, our method cannot produce plausible results, while the generated normals look reasonable, see Figure 9. We suspect it is due to the multi-view images being inconsistent across different views, as observed by many other works [24, 65].

VDMs are predominantly used for modeling organic shapes, yet the idea of modeling-by-parts can be applied to the majority of 3D shapes. There are exciting further avenues for part-based 3D generative models.

References

[1] Blender. vector displacement node. https://docs.blender.org/manual/en/latest/render/shader_nodes/vector/vector_displacement.html. 1

[2] Maya. vector displacement - arnold for maya. https://help.autodesk.com/view/ARNOL/ENU/?guid=arnold_for_maya_displacement_am_Vector_Displacement_html.

[3] Mudbox. vector displacement maps overview. https://download.autodesk.com/us/mudbox/help2011_5/index.html?url=./files/WS73099cc142f487552b5ac6c412649166e6e-6762.htm,topicNumber=d0e21754.

[4] Zbrush. vector displacement maps. https://help.maxon.net/zbr/en-us/Content/html/user-guide/3d-modeling/exporting-your-model/vector-displacement-maps/vector-displacement-maps.html. 1

[5] Adobe firefly. https://www.adobe.com/sensei/generative-ai/firefly.html. 6

[6] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2293–2303, 2019. 2

[7] Melinos Averkiou, Vladimir Kim, Youyi Zheng, and Niloy J. Mitra. Shapesynth: Parameterizing model collections for coupled shape exploration and synthesis. *Computer Graphics Forum (Special issue of Eurographics 2014)*, 2014. 3

[8] Gavin Barill, Nia Dickson, Ryan Schmidt, David I.W. Levin, and Alec Jacobson. Fast winding numbers for soups and clouds. *ACM Transactions on Graphics*, 2018. 5

[9] James F Blinn. Simulation of wrinkled surfaces. In *Seminal graphics: pioneering efforts that shaped the field*, pages 111–117. 1998. 2

[10] Edwin Earl Catmull. *A subdivision algorithm for computer display of curved surfaces*. The University of Utah, 1974. 2

[11] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2

[12] Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas Guibas, and Vladlen Koltun. Probabilistic reasoning for assembly-based 3d modeling. *ACM Trans. Graph.*, 30(4), 2011. 3

[13] Siddhartha Chaudhuri, Evangelos Kalogerakis, Stephen Giguere, and Thomas Funkhouser. AttribIt: Content creation with semantic attributes. *ACM Symposium on User Interface Software and Technology (UIST)*, 2013. 3

[14] Yun-Chun Chen, Selena Ling, Zhiqin Chen, Vladimir G. Kim, Matheus Gadelha, and Alec Jacobson. Text-guided controllable mesh refinement for interactive 3d modeling. In *Proceedings of the ACM SIGGRAPH*. ACM, 2024. 5

[15] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5939–5948, 2019. 2

[16] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 628–644. Springer, 2016. 2

[17] Robert L Cook. Shade trees. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 223–231, 1984. 2

[18] Robert L Cook, Loren Carpenter, and Edwin Catmull. The reyes image rendering architecture. *ACM SIGGRAPH Computer Graphics*, 21(4):95–102, 1987. 2

[19] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 2, 3, 5

[20] Slava Elizarov, Ciara Rowles, and Simon Donné. Geometry image diffusion: Fast and data-efficient text-to-3d with image-based surface representation. *arXiv preprint arXiv:2409.03718*, 2024. 2

[21] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. Modeling by example. In *ACM SIGGRAPH 2004 Papers*, 2004. 2

[22] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018. 2, 5

[23] Xianfeng Gu, Steven J Gortler, and Hugues Hoppe. Geometry images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 355–361, 2002. 2

[24] Junlin Han, Filippos Kokkinos, and Philip Torr. Vfusion3d: Learning scalable 3d generative models from video diffusion models. In *European Conference on Computer Vision*, pages 333–350. Springer, 2025. 2, 8

[25] Zexin He and Tengfei Wang. Openlrm: Open-source large reconstruction models. https://github.com/3DTopia/OpenLRM, 2023. 6

[26] Paul S Heckbert. Survey of texture mapping. *IEEE computer graphics and applications*, 6(11):56–67, 1986. 2

[27] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. In *The Twelfth International Conference on Learning Representations*. 2, 4, 6, 7, 8

[28] Arjun Jain, Thorsten Thormählen, Tobias Ritschel, and Hans-Peter Seidel. Exploring shape variations by 3d-model decomposition and part-based recombination. *Comput. Graph. Forum*, 31(2pt3):631–640, 2012. 3

[29] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 2

[30] Tomomichi Kaneko, Toshiyuki Takahei, Masahiko Inami, Naoki Kawakami, Yasuyuki Yanagida, Taro Maeda, and Susumu Tachi. Detailed shape representation with parallax mapping. In *Proceedings of ICAT*, pages 205–208, 2001. 2

[31] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013. 5

[32] Vladislav Kreavoy, Dan Julius, and Alla Sheffer. Model composition from interchangeable components. In *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, page 129–138, USA, 2007. IEEE Computer Society. 2

[33] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics (ToG)*, 39(6):1–14, 2020. 4

[34] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In *The Twelfth International Conference on Learning Representations*. 2, 4

[35] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. GRASS: Generative recursive autoencoders for shape structures. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 36(4), 2017. 3

[36] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10072–10083, 2024. 2

[37] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[38] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023. 2

[39] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. In *The Twelfth International Conference on Learning Representations*. 2

[40] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. *CVPR*, 2023. 2, 4, 6, 7, 8

[41] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. *arXiv preprint arXiv:2306.07279*, 2023. 5

[42] Tiange Luo, Justin Johnson, and Honglak Lee. View selection for 3d captioning via diffusion ranking. *arXiv preprint arXiv:2404.07984*, 2024. 5

[43] Nelson L Max. Horizon mapping: shadows for bump-mapped surfaces. *The Visual Computer*, 4:109–117, 1988. 2

[44] Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Realfusion: 360deg reconstruction of any object from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8446–8455, 2023. 2

[45] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 2

[46] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 2

[47] Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. Large steps in inverse rendering of geometry. *ACM Transactions on Graphics (TOG)*, 40(6):1–13, 2021. 4

[48] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 577–582. 2023. 5, 6

[49] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 2

[50] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2, 6

[51] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023. 2, 6, 7, 8

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 2, 6

[53] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 3

[54] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. `https://github.com/mseitzer/pytorch-fid`, 2020. Version 0.3.0. 2, 6

[55] Alla Sheffer, Bruno Levy, Maxim Mogilnitsky, and Alexander Bogomyakov. Abf++: Fast and robust angle based flattening. In *ACM Transactions on Graphics (TOG)*, pages 311–330. ACM, 2005. 4

[56] Chao-Hui Shen, Hongbo Fu, Kang Chen, and Shi-Min Hu. Structure recovery by part assembly. *ACM Trans. Graph.*, 31 (6), 2012. 3

[57] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023. 2, 3

[58] Zifan Shi, Sida Peng, Yinghao Xu, Yiyi Liao, and Yujun Shen. Deep generative models on 3d representations: A survey. *CoRR*, abs/2210.15663, 2022. 1

[59] Zhengwentai Sun. clip-score: Clip score for pytorch. https://github.com/taited/clip-score, 2023. 6

[60] Minhyuk Sung, Hao Su, Vladimir G Kim, Siddhartha Chaudhuri, and Leonidas Guibas. ComplementMe: Weakly-supervised component suggestions for 3D modeling. *ACM Transactions on Graphics (TOG)*, 36(6):226, 2017. 3

[61] László Szirmay-Kalos and Tamás Umenhoffer. Displacement mapping on the gpu—state of the art. In *Computer graphics forum*, pages 1567–1592. Wiley Online Library, 2008. 2

[62] Kenshi Takayama, Ryan Schmidt, Karan Singh, Takeo Igarashi, Tamy Boubekeur, and Olga Sorkine. GeoBrush: Interactive mesh geometry cloning. *Comput. Graph. For. (Proc. EUROGRAPHICS)*, 30(2):613–622, 2011. 3

[63] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22819–22829, 2023. 2

[64] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2025. 2, 4

[65] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In *European Conference on Computer Vision*, pages 439–457. Springer, 2025. 2, 8

[66] Lifeng Wang, Xi Wang, Xin Tong, Stephen Lin, Shimin Hu, Baining Guo, and Heung-Yeung Shum. View-dependent displacement mapping. *ACM Transactions on graphics (TOG)*, 22(3):334–339, 2003. 2

[67] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018. 2

[68] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. In *The Twelfth International Conference on Learning Representations*. 2, 4

[69] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023. 6

[70] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. Crm: Single image to 3d textured mesh with convolutional reconstruction model. *CoRR*, 2024. 2

[71] Daniel Watson, William Chan, Ricardo Martin Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. In *The Eleventh International Conference on Learning Representations*. 2

[72] Xinyue Wei, Kai Zhang, Sai Bi, Hao Tan, Fujun Luan, Valentin Deschaintre, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Meshlrm: Large reconstruction model for high-quality mesh. *arXiv preprint arXiv:2404.12385*, 2024. 2, 4

[73] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10130–10139, 2019. 5

[74] Desai Xie, Sai Bi, Zhixin Shu, Kai Zhang, Zexiang Xu, Yi Zhou, Sören Pirk, Arie Kaufman, Xin Sun, and Hao Tan. Lrm-zero: Training large reconstruction models with synthesized data. In *NeurIPS*, 2024. 2

[75] Xiaohua Xie, Kai Xu, Niloy J. Mitra, Daniel Cohen-Or, Wenyong Gong, Qi Su, and Baoquan Chen. Sketch-to-design: Context-based part assembly. *Computer Graphics Forum*, xx(xx):xx, 2013. 3

[76] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. 2

[77] Kai Xu, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. Fit and diverse: Set evolution for inspiring 3d shape galleries. *ACM Transactions on Graphics, (Proc. of SIGGRAPH 2012)*, 31(4):57:1–57:10, 2012. 3

[78] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *Advances in neural information processing systems*, 32, 2019. 2

[79] Xingguang Yan, Han-Hung Lee, Ziyu Wan, and Angel X Chang. An object is worth 64x64 pixels: Generating 3d object via image diffusion. *arXiv preprint arXiv:2408.03178*, 2024. 2

[80] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 601–610, 2020. 2

[81] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024. 2, 6, 8

[82] Lexing Ying, Aaron Hertzmann, Henning Biermann, and Denis Zorin. Texture and shape synthesis on surfaces. In *Rendering Techniques 2001: Proceedings of the Eurographics Workshop in London, United Kingdom, June 25–27, 2001 12*, pages 301–312. Springer, 2001. 2

[83] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021. 2

[84] Dan Zeng, Erin Cahmbers, David Letscher, and Tao Ju. To cut or to fill: a global optimization approach to topological simplification. *ACM Transactions on Graphics, (Proc. of SIGGRAPH 2020)*, 39(6):201:1–201:18, 2020. 4

[85] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–16, 2023. 2

[86] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pages 1–19. Springer, 2025. 2, 4

[87] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 43(4):1–20, 2024. 2

[88] Xin-Yang Zheng, Hao Pan, Yu-Xiao Guo, Xin Tong, and Yang Liu. Mvdˆ 2: Efficient multiview 3d reconstruction for multiview diffusion. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2