GenVDM: Generating Vector Displacement Maps From a Single Image

Supplementary Material

A. Details on Evaluation

For each VDM or Scalar DM, we apply it on a plane mesh and ensure that they are roughly of equal size with other comparison baselines. We add a plane behind each shape generated by single image to 3D reconstruction methods to ensure fair comparison. We render the resulting shapes in 13 different camera poses with (elevation angle, azimuth angle) $= (0°, \pm 60°)$, $(0°, \pm 45°)$, $(0°, \pm 30°)$, $(\pm 60°, 0°)$, $(\pm 45°, 0°)$, $(\pm 30°, 0°)$, $(0°, 0°)$ respectively. We use the default texture-less gray shading to render the shapes. For CLIP-similarity metric, we use ViT-B/32 model for evaluation. For 3D-FID score , we calculate the score between the set of rendered images and the set of input images for all shapes and use model checkpoint provided by [57]. We convert the input images into gray-scale images and add a gray square behind each input image to make its appearance align with that of the rendered images. See figure 11 for some example images used for evaluation.

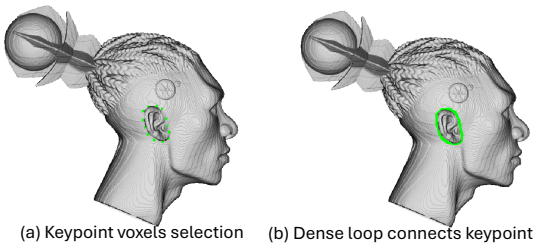
B. Details on Data Preparation

As shown in figure 10, Our 3D lasso tool is built upon voxel renderer. During annotation, we first select a few keypoint voxels to form a sparse loop around the region of interest, and then we find the dense loop by finding a shortest voxel path on the voxel surface that connects these selected keypoint voxels. To extract the segmented part, we remove voxels of the dense loop and use a flooding algorithm to identify the region enclosed by the annotated voxel loop. We then sample densely on the sub-mesh that is contained by the selected voxel region to obtain point clouds with normals for surface reconstruction. Since the sub-mesh may contain triangles that are not on the surface of the shape, we use fast winding number [8, 28] to remove interior points. We then use Screened Poisson Surface Reconstruction implemented in Open3D[92] to remesh and obtain a single connected surface. We also filter out VDM shapes of poor quality after the data preparation pipeline to enhance data quality.

C. Details on VDM Reconstruction

Our neural deformation field network consists of an 8-layer MLP of latent dimension 512 with residual connection at the fourth layer. We use LeakyReLU[45] as activation function and set negative slope to 0.01. We use Adam[33] optimizer and set learning rate to 5e-4 to optimize 3000 epochs. Specifically, we first initialize the MLP so that the initial output points form a 3D square plane. We achieve this by optimizing against an initialization optimization objective:

$$\underset{\theta}{\operatorname{argmin}} \quad \frac{1}{|P|} \sum_{p \in P} \|\phi_\theta(p) - \text{proj}(p)\|_2^2$$



(a) Keypoint voxels selection (b) Dense loop connects keypoint voxels

Figure 10. Our 3D lasso tool for segmentation. (a) We first select keypoint voxels around interested region. (b) We then find dense voxel loop by connecting keypoint voxels by shortest path.

Method	CLIPImg↑	CLIPText↑	3D-FID↓
Wonder3D [42]	0.8143	0.2504	208.7
Magic123 [54]	0.8241	0.2488	218.5
LRM [27]	0.8122	0.2472	246.3
Scalar DM	0.8201	0.2529	223.4
Ours	**0.8460**	**0.2639**	**197.3**

Table 3. Additional quantitative comparison with baselines.

where P are sets of sampled points from $[0, 1]^2$ and $\text{proj}(p)$ maps p to the corresponding 3D point in a pre-defined 3D square plane. We then use the optimization objective proposed in 3.2 for subsequent optimization. For mesh optimization comparison method, we set the laplacian regularization loss ratio to 1e-4 compared with chamfer reconstruction loss. We optimize for 1200 iterations and no remeshing is done during the optimization process. For topological fixing and tutte embedding comparison method, we use the built-in parametrization algorithm in Blender[1] "Unwrap" function in UV editing after fixing boundary vertices.

D. Details on Training

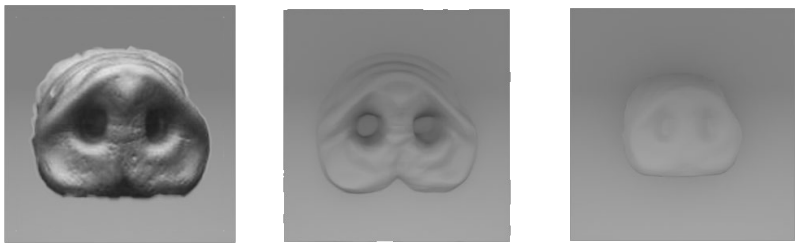
We finetune our multi-view normal generation model on the checkpoint provided by Zero123++[60] on 8 NVIDIA A100 GPUs for 3 days. We finetune it with a base learning rate of 1e-5 and dropout condition probability of 0.1. We set the batch size to 48 and optimized for 50000 steps. We do not use gradient accumulation.

E. Additional Evaluation

We added 50 RGB images from the Internet and a text-to-image model [5] into our testing dataset of 50 images used in the paper. Quantitative results on the new 100-image dataset are shown in Table 3.

F. More Result

We present more results generated by our model for further qualitative evaluation.



(a) Input image (b) Our method image (c) Comparison baseline image

Figure 11. Example images used for computing quantitative results. (a) Input image. (b) Rendered image of our method. (c) Rendered image of a comparison baseline method, Wonder3D[42].

Figure 12. More results.