Hi3DGen: High-fidelity 3D Geometry Generation from Images via Normal Bridging

Supplementary Material

6. More Details for the Method

Dataset Details Our training pipeline utilizes carefully curated datasets for both image-to-normal and normal-to-geometry stages. For image-to-normal training, we employ two complementary datasets: a diverse realistic dataset constructed following the methodology of Depth-pro [4], and a large-scale synthetic dataset comprising 20M RGB-to-normal pairs generated by rendering 40 images per asset from our collection of 500K DetailVerse assets.

For the normal-to-geometry training stage, we construct a comprehensive dataset containing 170K cleaned 3D assets sourced from Objaverse [13] and 700K synthesized 3D assets from our DetailVerse. Each asset is rendered from 40 different viewpoints following the rendering protocol established in Trellis [74], ensuring diverse viewing angles and lighting conditions.

To evaluate the generalization capability of our image-to-normal estimator on real-world scenarios, we validate performance on the LUCES-MV [41] reconstruction dataset, which provides challenging multi-view reconstruction scenarios. For comprehensive visual comparisons and user studies presented in this work, we collected evaluation images from multiple sources including the Hyper3D website [12], Hunyuan3D-2.0 project page [59], and Dora project page [54], ensuring diverse and representative test cases across different domains and complexity levels.

Implementation Details For image-to-normal, we adopt GenPercept [77] architecture for Normal Regression network. We initialize the encoder and decoder weights from the Stable Diffusion V2.1 [53], finetuned using the AdamW optimizer with a fixed learning rate of 3×10^{-5}. For normal-to-geometry, we build upon the Trellis [74], incorporating classifier-free guidance (CFG) [26] with a drop rate of 0.1 and AdamW [43] optimizer with a fixed learning rate of 1×10^{-4}. We implement noise injection using an EDM-style noise sampler, which randomly adds noise to the encoder output latents before they are input to the decoder. Specifically, we follow EDM to use standard parameters with $\sigma_{\min} = 0.002$ and $\sigma_{\max} = 80.0$. We guarantee the SNR of the features by selecting timestamps from 0 to 400, which we empirically found maintains coarse shape knowledge — this approach aligns with Instruct-Pix2Pix, which also adds noise in a middle timerange to avoid structural changes. To better preserve coarse knowledge while instructing the encoder to focus on detailed information, we follow ControlNet [83] to add a secondary encoder by copying the weights of the SD2.1 encoder and concatenating multi-layer features with the decoder for dual-stream training. Specifically, an image is first processed by the VAE before entering the two encoders. The encoders do not share weights since they are designed to learn different frequency information from the images. For the normal-to-geometry training stage, we finetune the Large variant of Trellis using 8 NVIDIA A800 GPUs (80GB each) for 50k steps with a batch size of 256. During inference, we set the CFG strength to 3.0 and use 50 sampling steps to achieve optimal results.

Training Details For training our I2N method, we input identical image latents to the dual encoders, where the coarse encoder remains noise-free with no modifications to any layers, while for the fine-grained encoder, we follow the approach described above to inject noise on the encoder output. Notably, we feed the noised features to the decoder layers rather than back to the encoder layers. For domain-specific training, we first train on real-world data from the Depth-pro dataset for 50,000 steps with a batch size of 256 at 768px resolution. We randomly crop images at varying aspect ratios before resizing to 768px. Subsequently, we train our model on rendered images from DetailVerse and Objaverse[13]. For DetailVerse, we render 40 spherical views per object using nvdiffrast, varying the radius and field of view. For Objaverse training, we use the 40-view renders from GObjaverse, applying the filter criteria from RichDreamer[49] to select 170K high-quality samples. We fine-tune in this second stage on the synthetic dataset while freezing the coarse encoder. For training our N2G method, we follow Trellis to employ rectified flow for model fine-tuning. We reuse the Sparse Structure VAE and Structured Latent VAE without modification, as DetailVerse is generated using Trellis (ensuring domain compatibility), and our selected Objaverse subset is already included in the original Trellis training dataset.

Inference Details During inference of Hi3DGen, we first utilize an off-the-shelf background removal model to isolate the foreground object. We crop the foreground and pad the image to a square format, then resize it to 768×768 resolution before inputting it to our NiRNE model. During the inference of NiRNE, we do not inject any noise into the encoder features to ensure stable inference and maximize the preservation of detail information captured by the fine-grained encoder. Given the estimated normal map, we set the background to white and input the normal map to NoRLD. Trellis employs a two-stage generation pipeline to

Table S4. Image-to-Normal estimation evaluation on Luces-MV (SNE). Comparisons of NiNRE with SOTA photometric stereo techniques. **Bold** indicates the second best results and **Red** indicates best results.

Method	Bowl	Buddha	Bunny	Cup	Die	Hippo	House	Owl	Queen	Squirrel	Ave.
SDM-UniPS (K=2)	37.65	26.24	**29.02**	23.70	**26.32**	31.45	40.68	**24.56**	27.14	26.10	29.286
SDM-UniPS (K=4)	**31.64**	**20.59**	**23.23**	**23.39**	**25.58**	**21.91**	**38.61**	**22.26**	**25.97**	**24.04**	**25.722**
Ours	**34.55**	**21.13**	30.45	**17.47**	27.20	**24.64**	**34.58**	25.15	**26.82**	**24.29**	**26.628**

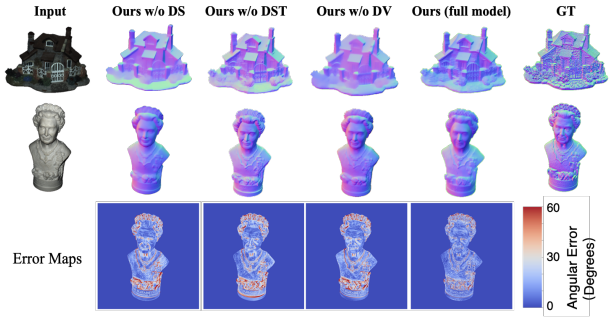

Figure S11. Ablations on image-to-normal estimation.

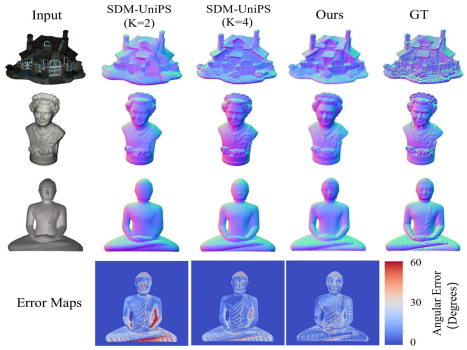

Figure S12. Qualitative comparison of image-to-normal estimation with SOTA Photometric Stereo-based Method, SDM-UniPS.

produce structured latents, which first generates the sparse structure, followed by the local latents attached to it. Following the same approach as Trellis, we first generate the sparse structure represented by sparse voxels, then initialize noise on the sparse voxel representation to generate the final structure latents using our fine-tuned structure latents flow model. The final mesh is generated using the pre-trained mesh decoder.

Metric Explanation For comprehensive evaluation of image-to-normal, we adopt metrics from Dora [10] to quantify normal map accuracy, with particular emphasis on sharp edges where geometric details are most salient. Specifically, we compute the Sharp Normal Error (SNE) through a three-step process: Firstly, we detect salient regions in the ground truth normal maps through canny. Secondly, we dilate these masked regions to ensure complete coverage of edge features. Finally, we calculate the normal angle error within these masked regions. For completeness and fair comparison with existing methods, we also report the Normal Error (NE) across the entire normal map, measured in degrees. For evaluating normal-to-geometry conversion, we render normal maps from 22 fixed, evenly spaced viewpoints around each object using nvdiffrast [36], which is used to compute SNE and NE.

7. More Details for the DetailVerse

To ensure the quality of our synthesized meshes, we implement a rigorous multi-stage data generation and filtering pipeline that combines expert evaluation with automated assessment techniques.

Step 1: Semantic Text Prompt Curation We initiate the 3D data synthesis process with text prompts rather than image prompts, as textual descriptions enable more precise control over semantic diversity, thereby ensuring variety in the resulting geometries. To collect high-quality text prompts with semantic diversity, we first sourced approximately 14M raw prompts from DiffusionDB [67], covering a wide range of topics relevant to AI generation applications. We employed a LLaMA-3-8B model [60], fine-tuned with manually annotated examples, to categorize these prompts into four distinct classes: (i) Single Objects; (ii) Multiple Objects; (iii) Scenes; and (iv) Others. Only prompts from classes (i) and (ii) were retained, yielding approximately 1M high-fidelity prompt candidates.

Next, we applied rule-based filtering to preserve geometric and semantic attributes while eliminating stylistic modifiers. Empirically, we observed that input images with near-isometric viewpoints and CGI-rendered aesthetics significantly enhance the fidelity of 3D synthesis. Thus, we implemented structural prompt standardization to prompting the image generation. Specifically, we applying domain-specific prompt templates to enforce explicit geometric cues and structural clarity (e.g., "isometric perspective", "Unreal Engine 5 Rendering", "4K", "MasterPiece"). This comprehensive process yielded approximately 1.5 million well-curated and natural prompts.

Step 2: High-Quality Image Generation With our diverse text prompt collection established, the next step involved generating corresponding images suitable for 3D asset synthesis. The key requirements for these images were: (i) high visual fidelity with rich details that accurately reflect the textual descriptions; and (ii) specific viewpoints and styles that facilitate robust 3D reconstruction.

We integrated the state-of-the-art Flux.1-Dev [35] as our image generator. To ensure detailed output, we filtered the generated images by ranking their sharpness according to the number of sharp pixels, as calculated using Canny edge detection, and retained only the top 50%. For each prompt, we randomly selected a seed to encourage variety, generating exactly one image per prompt.

To mitigate geometry distortion in the resulting 3D models, we utilized OrientAnything [66], a robust object orientation estimation model, to measure the alignment between the camera view and canonical object orientation. Images with angular deviations exceeding 60° were rejected to prevent structural distortions and preserve geometric fidelity. Through this filtering process, we preserved 1 million high-quality images for the subsequent 3D synthesis stage.

Step 3: Robust Image-to-3D Synthesis We employed Trellis [74], a state-of-the-art two-stage 3D generator, to produce high-fidelity 3D objects from the prepared images. Given its superior performance with high-quality inputs, we initially generated a set of preliminary meshes.

To ensure mesh quality, we implemented a rigorous data cleaning process combining expert evaluation with automated assessment. We randomly sampled 10K meshes and engaged 10 trained experts to conduct triple-blind quality assessments. The evaluation criteria primarily focused on surface quality, specifically examining whether the rendered normal maps contained holes or noise artifacts.

Based on these expert annotations, we trained a quality assessment network using DINOv2 [45] features. Specifically, we extracted features from four equiangular rendered normal maps of each mesh and trained a three-layer MLP classifier for quality scoring. This trained network was then applied to evaluate the entire dataset. Models that received positive classifications across all four views were selected for training our NoRLD model. Through this comprehensive quality assurance process, we retained 700K high-quality object meshes to form our DetailVerse dataset. A data gallery is shown in Fig. S13, and better visualizations are presented in the demo video.

8. More Ablation Studies

NiRNE Ablation We provide qualitative results to supplement the ablation studies on the proposed NiRNE. As shown in Fig. S11, each component makes positive role in the final performance.

9. More Results

More Image-to-Normal Results We compare NiRNE with SOTA photometric stereo technique (SDM-UniPS [30]), which works in a different setup that requires input images under K different lightning conditions. (As shown in Fig. S12).

More Comparisons We give more qualitative comparisons in Fig. S14, which shows our normal-bridged Hi3DGen can achieve more consistent 3D detailed geometries with input images than existing methods. Better visualizations are presented in the demo video.

Figure S13. More DetailVerse data exhibition.

Figure S14. More 3D generation results comparison.