MAGE : Single Image to Material-Aware 3D via the Multi-View G-Buffer Estimation Model

Supplementary Material

This supplementary material provides additional information and experiment results of the main paper, "MAGE: Single Image to Material-Aware 3D via the Multi-View G-Buffer Estimation Model", including detailed descriptions of the implementation details and more visual results to complement the experiments reported in the main paper.

1. More Visual Results of MAGE

In this section, we present additional visual results of MAGE in both 2D multi-view G-buffers estimation and dynamic 3D relighting.

2D Multi-view G-buffers Estimation Results. To demonstrate the generalization ability of MAGE for estimating multi-view G-buffers from wild images, we show the multi-view RGB images and G-buffers generated by MAGE based on a single AI-created image (Fig. 1) and a real captured image (Fig. 2). As can be seen, our method generates consistent multi-view RGB images and multi-view G-buffers for both AI-created and real captured images, indicating the robustness of MAGE to out-of-training-data images.

Dynamic 3D Relighting Video Results To better demonstrate the quality of our reconstruction results, we provide supplementary video sequences showing the dynamic relighting of the teaser results, where the environment lighting rotates around the z-axis. In the video, our results demonstrate a visually pleasing, natural, and photorealistic appearance, while MeshFormer [4] tends to produce noticeable lighting inconsistencies and artifacts under dynamic illumination conditions.

2. Implementation Details

In this section, we elaborate on the implementation details, including the training specifics of our G-buffer estimation network, the details of the implementation of the lighting response loss, and the sparse-view 3D reconstruction methodology used to convert the multi-view G-buffers into a final 3D object with material properties.

Training Details. As described in Sec.3.2 of the main paper, we initialize the G-buffer estimation network from the weights of Zero123++ [5]. Then, we transform the multi-step denoising U-Net into a deterministic single-step U-Net by (1) replacing the latent Gaussian noise with the latent representation of the tiled input RGB image in a 3×2 grid and (2) removing timestep sampling at each training step and fixing the timestep to $T = 999$, similar to [1]. The input image is randomly resized in the range of $[256 \times 256, 512 \times 512]$ to adapt to input images with various resolutions during inference. The output G-buffer is a tiled image with a size of 960×640 for each view, where each G-buffer component is supervised by a ground truth image with a size of 320×320. We train the G-buffer estimation network on our synthetic dataset using image space losses instead of latent space loss. The network is trained for 10K steps with a batch size of 2 and accumulated gradient steps of 2 on 4×H100 GPUs, which equals a total batch size of 16. The loss weights $\lambda_L, \lambda_X, \lambda_N, \lambda_A, \lambda_R, \lambda_M$ in Eq.8 of the paper are set to $10.0, 0.5, 0.5, 1.0, 3.0, 3.0$, respectively. The learning rate is set to 1×10^{-5}, and the training takes about 20 hours.

Details of Lighting Response Loss Our lighting response loss leverages the split-sum approximation from real-time rendering for efficient physically-based re-rendering. Following [2], the lighting integral is approximated as illustrated in Eq. 3 in the main paper, implemented through a hierarchical cube map structure. The base environment map is a high-resolution (typically 6×512×512) cube map with trainable parameters for each texel. To handle different material roughness levels efficiently, we maintain a chain of filtered mipmap levels generated through average pooling, with roughness clamped between 0.08 and 0.5 for stable training.

The filtering process is fully differentiable and consists of two main components. We use the lowest resolution mipmap level for diffuse lighting with a cosine-weighted integration over the hemisphere. Each mipmap level corresponds to pre-filtered environment lighting for specular reflections to increase roughness values. This creates a sequence of increasingly blurred environment maps that approximate the integration of the GGX normal distribution function for different roughness values.

During shading, the diffuse component is evaluated by sampling the diffuse cube map using the surface normal. The specular component combines multiple terms: view-dependent reflection vector, a pre-computed BSDF lookup table built using the view angle, and roughness to obtain the Fresnel-geometry term. Based on the material roughness, the specular environment lookup uses tri-linear filtering between appropriate mipmap levels. The final color combines both diffuse and specular terms, taking into account the material's metallic parameter which determines the ratio between specular and diffuse reflection. This implementation enables efficient all-frequency lighting esti-

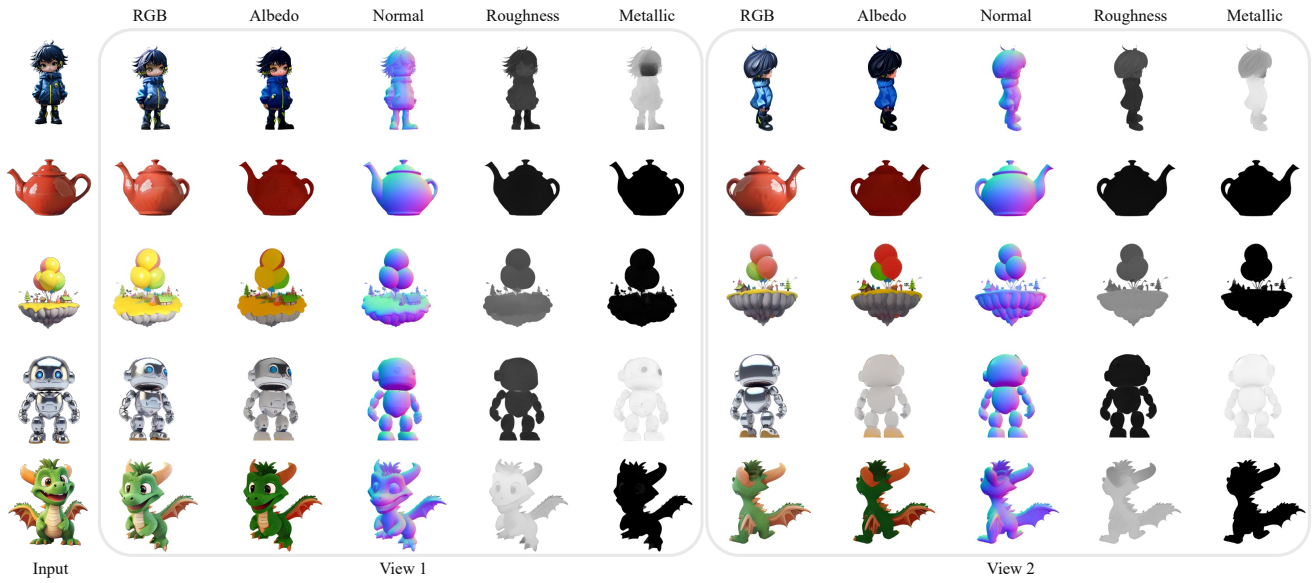

Figure 1. Multi-view RGB images and G-buffers generated by MAGE. All inputs are AI-created images. We only show two novel views here, while MAGE generates six novel views at inference.

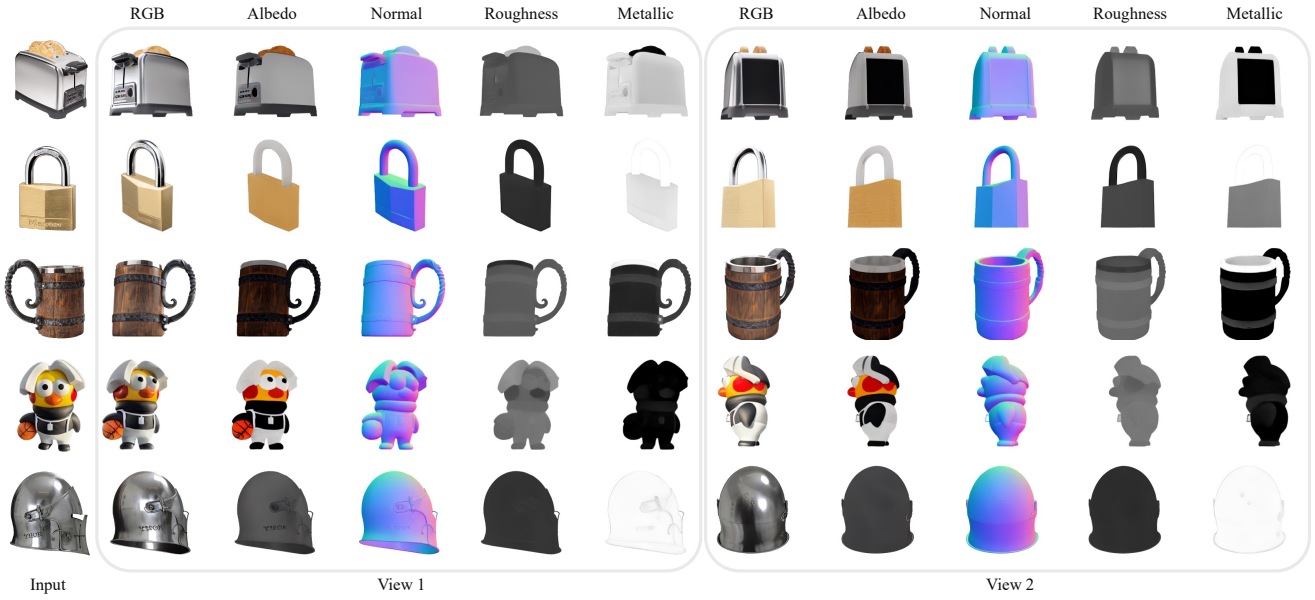

Figure 2. Multi-view RGB images and G-buffers generated by MAGE. All inputs are real captured images. We only show two novel views here, while MAGE generates six novel views at inference.

mation while maintaining compatibility with standard real-time rendering pipelines and physically based materials.

Sparse-View 3D Reconstruction. For sparse-view 3D reconstruction, we use Nvdiffrast [3] for mesh optimization from G-buffers. Our sparse-view 3D reconstruction pipeline takes as input the multi-view G-buffers predicted by our network, which are arranged in a $n \times 6$ grid containing n views and six domains G-buffers (RGB, normal map, depth, albedo, roughness, and metallic) for each view. We first obtain an initial mesh by converting the predictions to

a visual hull using known camera parameters and masks extracted from images. The core of our reconstruction process utilizes Nvdiffrast as the differentiable renderer to optimize both mesh vertices and texture coordinates. The optimization objective combines multiple supervision signals from our predicted G-buffers, including RGB appearance matching supervised by albedo, roughness, and metallic maps and a normal / depth alignment supervised by normal and depth maps. We implemented a normal / depth renderer for normal and depth alignment. Specifically, we minimize the difference between the rendered attributes and our predicted G-buffers through differentiable rendering to optimize.

We employ Xatlas to automatically generate UV parameterization for the optimized mesh for the material assignment. We project the predicted G-buffer attributes (albedo, roughness, metallic) onto the UV space to obtain the final ready-to-use 3D meshes with PBR materials in obj format. Using the angle-weighted averaging method, we blend multiple view contributions to handle view-dependent effects and potential inconsistencies across views. The final material properties are stored as 1024×1024 resolution UV maps for preserving fine details from our G-buffer predictions.

References

[1] Gonzalo Martin Garcia, Karim Abou Zeid, Christian Schmidt, Daan de Geus, Alexander Hermans, and Bastian Leibe. Fine-tuning image-conditional diffusion models is easier than you think. *arXiv preprint arXiv:2409.11355*, 2024. 1

[2] Brian Karis and Epic Games. Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice*, 4(3):1, 2013. 1

[3] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020. 2

[4] Minghua Liu, Chong Zeng, Xinyue Wei, Ruoxi Shi, Linghao Chen, Chao Xu, Mengqi Zhang, Zhaoning Wang, Xiaoshuai Zhang, Isabella Liu, Hongzhi Wu, and Hao Su. Meshformer: High-quality mesh generation with 3d-guided reconstruction model. *arXiv preprint arXiv:2408.10198*, 2024. 1

[5] 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. 1