

MeTTA: Single-View to 3D Textured Mesh Reconstruction with Test-Time Adaptation

Kim Yu-Ji¹

ugkim@postech.ac.kr

Hyunwoo Ha²

hyunwooaha@postech.ac.kr

Kim Youwang²

youwang.kim@postech.ac.kr

Jaeheung Surh³

jh.surh@bucketplace.net

Hyowon Ha^{3,2}

hyowon.ha@bucketplace.net

Tae-Hyun Oh^{1,2,4,†}

taehyun@postech.ac.kr

¹ Grad. School of AI

POSTECH, South Korea

² Dept. of Electrical Engineering

POSTECH, South Korea

³ Bucketplace, Co., Ltd., South Korea

⁴ Institute for Convergence

Research and Education

in Advanced Technology

Yonsei University, South Korea

Contents

1	Technical Details	2
1.1	Implementation Details	2
1.2	Texture Modeling	2
2	Additional Quantitative Analysis	3
2.1	Cross-domain Comparison	3
2.2	In-domain Comparison	4
3	Additional Qualitative Analysis	4
3.1	In-domain Comparison	4
3.2	Cross-domain Comparison	9
4	In-depth Analysis of Limitation and Discussion	11
4.1	Failure Cases	11
4.2	Discussion	11

Supplementary Material

This supplementary material presents technical details, analyses, and experiments not included in the main paper due to the space limit.

1 Technical Details

This section provides detailed information on the implementation details of the overall pipeline and physically-based rendering (PBR) modeling in the main paper.

1.1 Implementation Details

Experimental details. We use AdamW optimizer with gradient clipping and the respective learning rates of 1×10^{-3} for geometry and 1×10^{-3} for texture and optimize them simultaneously. We randomly sample 8 camera viewpoints for each iteration for rendering the novel views. We conduct training with one NVIDIA A6000 GPU for about 30 minutes. We leverage Open3D [23] to deal with SDF and point cloud representations.

Chamfer Distance. We measure Chamfer Distance to assess the quality of the mesh reconstruction. Point clouds are normalized in scale and aligned to the ground-truth point clouds by the iterative closest point (ICP) algorithm. 10K points are sampled for evaluating each mesh.

Image-to-3D module. We require a learning-based feed-forward mesh prediction stage employing the Image-to-3D module to obtain a preliminary coarse mesh and initial viewpoint of the input image. The Image-to-3D module encompasses various techniques capable of predicting a coarse mesh and an approximate viewpoint for the input image, *e.g.*, [16, 22].

Segmentation module. MeTTA harnesses the multi-view diffusion model [10] fine-tuned on large-scale synthetic datasets [4, 5], specifically designed for object rendering against a white background. Achieving precise object segmentation is pivotal for effectively leveraging the multi-view diffusion model, biased towards images with segmented white backgrounds. To automate the process of obtaining high-quality segmentation results, we make use of the latest segmentation models [4, 5]. While these models offer substantial automation, they still require some level of user-interactive querying. In response, we have integrated a grounding method [10] to obtain appropriate object detection as a query. Based on the detection results as a user-given query, we subsequently employ a user-interactive segmentation method to finalize the fine-grained segmentation results.

1.2 Texture Modeling

As explained in Section 3.5. of the main paper, we adopt physically-based rendering (PBR) material modeling [13] to optimize neural texture optimization. By employing PBR material modeling, we can achieve a realistic appearance for the reconstructed object and easily integrate it with various graphics engines (*e.g.*, Blender [3]) for practical applications. The PBR material properties, denoted as \mathbf{k}_{PBR} , consist of three fundamental elements: diffuse lobe parameters $\mathbf{k}_d \in \mathbb{R}^3$, the roughness and metalness term $\mathbf{k}_{rm} \in \mathbb{R}^2$, and the normal variation term $\mathbf{k}_n \in \mathbb{R}^3$. \mathbf{k}_{rm} consists of the roughness r and metalness term m . The first term, r , is a parameter of GGX [20] normal distribution function and affects how the material's

surface reflects light. The second term, m , is used with diffuse value \mathbf{k}_d for computing the specular term $\mathbf{k}_s = (1 - m) \cdot 0.04 + m \cdot \mathbf{k}_d$. We employ a tangent space normal map, denoted as \mathbf{k}_n , to capture intricate high-frequency lighting details on the surface. With a given scene environment light [14], we can compute a basic rendering equation as a basic image-based lighting model denoted by:

$$L_\theta(\mathbf{p}, \mathbf{c}) = \int_{\Omega} L_i(\mathbf{p}, \mathbf{c}_i) f_\theta(\mathbf{p}, \mathbf{c}_i, \mathbf{c})(\mathbf{c}_i \cdot \mathbf{n}_p) d\mathbf{c}_i, \quad (1)$$

where L is the rendered pixel color along the view direction \mathbf{c} of the 3D mesh surface point \mathbf{p} . L_i is the incident light from the given off-the-shelf environment map, and Ω is a hemisphere surrounding the surface with the altered surface normal \mathbf{n}_p . Additionally, $f_\theta(\mathbf{p}, \mathbf{c}_i, \mathbf{c})$ is the bidirectional reflectance distribution function (BRDF) modeled by PBR material modeling, \mathbf{k}_d , \mathbf{k}_{rm} , and \mathbf{k}_n . We can split Eq. 1 into diffuse term L_d and the specular term L_s as:

$$\begin{aligned} L(\mathbf{p}, \mathbf{c}) &= L_d(\mathbf{p}) + L_s(\mathbf{p}, \mathbf{c}), \\ L_d(\mathbf{p}) &= \mathbf{k}_d(1 - m) \int_{\Omega} L_i(\mathbf{p}, \mathbf{c}_i)(\mathbf{c}_i \cdot \mathbf{n}_p) d\mathbf{c}_i, \\ L_s(\mathbf{p}, \mathbf{c}) &= \int_{\Omega} \frac{DFG}{4(\mathbf{c} \cdot \mathbf{n}_p)(\mathbf{c}_i \cdot \mathbf{n}_p)} L_i(\mathbf{p}, \mathbf{c}_i)(\mathbf{c}_i \cdot \mathbf{n}_p) d\mathbf{c}_i, \end{aligned} \quad (2)$$

where D, F, and G indicate GGX (*i.e.*, microfacet) distribution, Fresnel term, and statistical light-blocking function, respectively. Following [14, 15], the split-sum approximation is used to calculate hemisphere integration. By merging the pixel colors in the rendered image along the view direction \mathbf{c} , we obtain the rendered image \mathbf{x} , representing the result of the rendering process, denoted as:

$$\mathbf{x} = R(\theta, \mathbf{c}), \quad (3)$$

where R refers to the differentiable renderer [9] and θ is the parameters of the MLP network that predict PBR material properties, as depicted in the main paper. We employ xatlas [20] for the generation of UV texture maps. As discussed in [14], the integration of sampled 2D textures directly into real graphics engines leads to the emergence of texture seams.

2 Additional Quantitative Analysis

In this section, we provide further quantitative comparisons in both cross-domain and in-domain scenarios. We evaluate cross-domain performance on a subset of the 3D-Front dataset [6] and in-domain performance on a subset of the Pix3D dataset [13], both of which contain ground-truth 3D meshes.

2.1 Cross-domain Comparison

In this section, we provide quantitative comparisons for cross-domain image to shape reconstruction. We compare the same samples in Table. 2 in the main paper. For cross-domain comparison, we train all methods, excluding our model MeTTA, on Pix3D [13]. Then, all methods evaluate on 3D-Front [6]. MeTTA shows comparable geometry reconstruction with previous methods, especially in the Chamfer Distance (See Table S1). It is noteworthy that we do not employ any 3D mesh data in our test-time optimization process.

Metric	MGN [10]	LIEN [12]	InstPIFu [10]	SSR [9]	MeTTA (Ours)
Chamfer Distance ↓	0.1089	0.0975	0.0992	0.1948	0.0943
F-Score (%) ↑	27.32	34.29	31.65	16.51	29.96

Table S1: **Cross-domain evaluation of feed-forward methods.** We measure the Chamfer Distance and F-Score between the predicted and ground-truth meshes. We conduct the experiment to show the test-time adaptation ability of the unseen test dataset, 3D-Front [9]. Note that although we utilize the icp algorithm, the result of SSR [9] could have unexpected errors due to its rotated and translated output geometry results.

Metric	MGN [10]	LIEN [12]	InstPIFu [10]	SSR [9]	MeTTA (Ours)
Chamfer Distance ↓	0.0494	0.0319	0.0825	0.1528	0.0612
F-Score (%) ↑	60.75	81.01	60.75	24.28	45.48

Table S2: **In-domain evaluation of feed-forward methods.** We measure Chamfer Distance and F-Score between the predicted and ground-truth meshes. We conduct the experiment to show the test-time adaptation ability of Pix3D [18], which is countered when training the Image-to-3D module. Note that although we utilize the ICP algorithm, the result of SSR [9] could have unexpected errors due to its rotated and translated output geometry results.

2.2 In-domain Comparison

We also evaluate our 3D object mesh reconstruction quality at the in-domain scenarios. Note that our optimization process does not access the ground-truth 3D information, *e.g.*, point clouds, voxels, and meshes, while previous methods [9, 10, 12, 12] are directly trained with Chamfer Distance with ground-truth meshes as supervision. Despite this, as shown in Table S2, MeTTA shows comparable geometry reconstruction with others. It is worth noticing that our method also reconstructs image-aligned geometry with realistic textures, whereas others are limited in reconstructing only 3D geometry even trained with 3D shape dataset [18].

3 Additional Qualitative Analysis

This section presents additional qualitative analyses due to space constraints in the main paper. We provide visual results for both in-domain scenarios on the Pix3D dataset [18], as well as the 3D-Front dataset [9] and real scenes.

3.1 In-domain Comparison

We assess the performance of our method on the Pix3D dataset, which aligns with our in-domain distribution, resulting in favorable initial mesh predictions as shown in Figs. S1, S2, S3 and S4. However, there are instances of erroneous predictions, which our approach effectively rectifies, enhancing the realistic appearance of the reconstruction results. It is important to note that changes in brightness and contrast may occur due to variations in lighting intensity (*i.e.*, different environment maps).



Figure S1: **Additional in-domain experiments about Pix3D [18].** We showcase the effectiveness of our test-time adaptation in in-domain scenarios. Even in the in-domain settings, the initial mesh prediction is inaccurate with no textures. With our test-time adaptation process, we show that fine-grained geometry with realistic textures.



Figure S2: **Additional in-domain experiments about Pix3D [18].** We showcase the effectiveness of our test-time adaptation in in-domain scenarios. Even in the in-domain settings, the initial mesh prediction is inaccurate with no textures. With our test-time adaptation process, we show that fine-grained geometry with realistic textures.

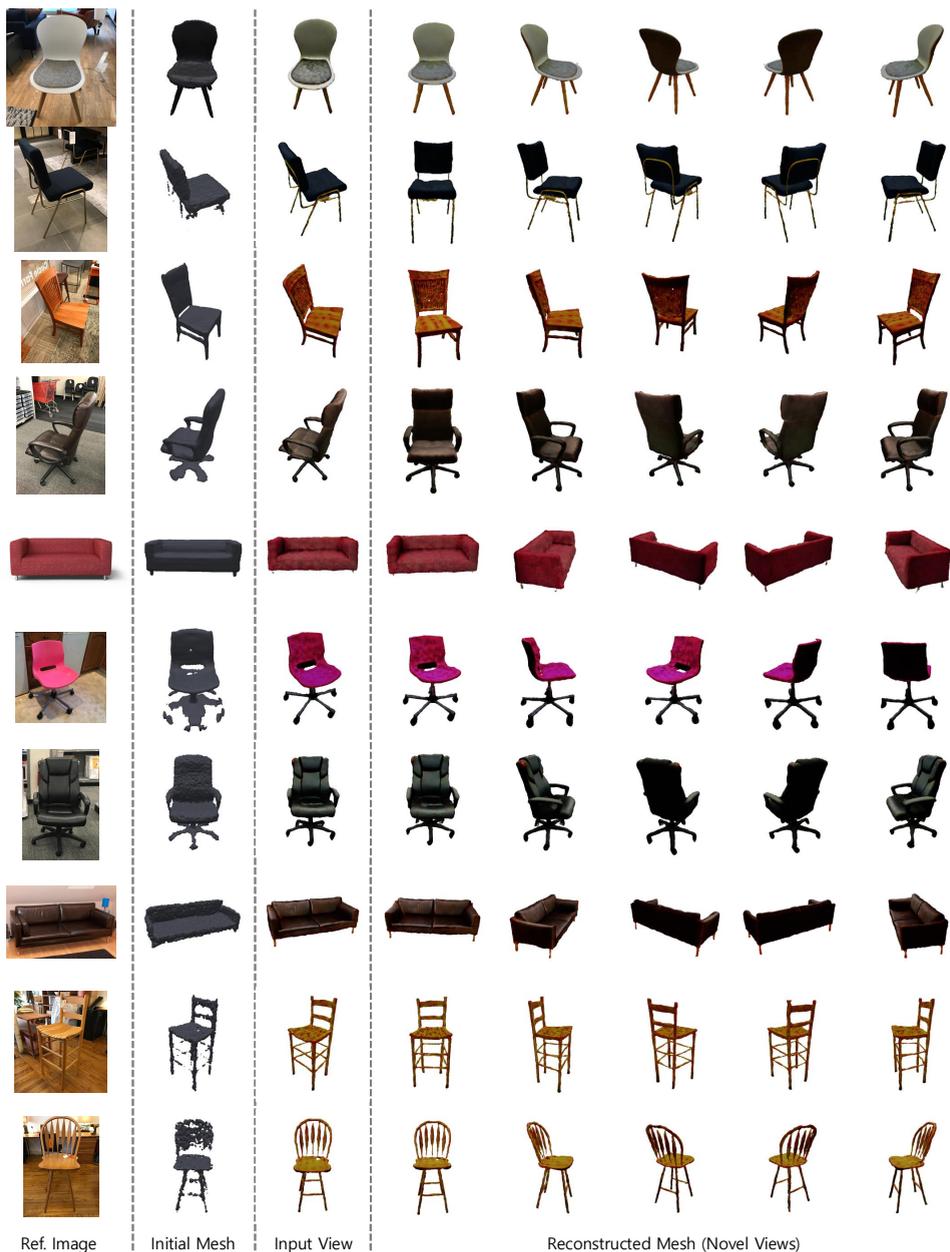


Figure S3: **Additional in-domain experiments about Pix3D [18].** We showcase the effectiveness of our test-time adaptation in in-domain scenarios. Even in the in-domain settings, the initial mesh prediction is inaccurate with no textures. With our test-time adaptation process, we show that fine-grained geometry with realistic textures.



Figure S4: **Additional in-domain experiments about Pix3D [18].** We showcase the effectiveness of our test-time adaptation in in-domain scenarios. Even in the in-domain settings, the initial mesh prediction is inaccurate with no textures. With our test-time adaptation process, we show that fine-grained geometry with realistic textures.



Figure S5: **Additional unseen real-world experiments.** We show the additional unseen real-world, *i.e.*, cross-domain experiments with the dataset which we manually acquired.

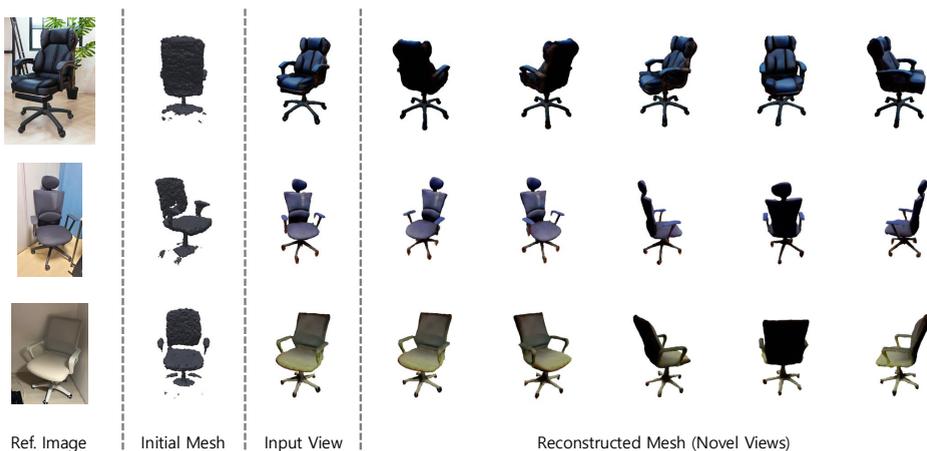


Figure S6: **Additional unseen in-the-wild experiments.** We show the additional in-the-wild, *i.e.*, cross-domain experiments with the dataset we acquired from the web.

3.2 Cross-domain Comparison

We evaluate the performance of an input image from previously unseen distributions through a real scene dataset that we directly acquired and an in-the-wild dataset from the web. As depicted in Figs. S5 and S6, real-world scenarios represent entirely new domains of images that we have not encountered before. Consequently, initial mesh predictions struggle to reflect the object shapes within the input image accurately. However, our test-time adaptation method enables us to obtain fine-grained textured meshes that not only capture the geometry of the input images but also incorporate their textures.

We demonstrate the effectiveness of our method on the 3D-Front [6] dataset, which represents an unseen cross-domain distribution, as illustrated in Fig. S7. These samples fall outside the training distribution and have not been encountered during training, so initial mesh predictions may not align well with the input image objects. However, through our test-time adaptation approach, we can successfully reconstruct object shapes and textures.



Figure S7: **Additional cross-domain experiments about 3D-Front [16].** We showcase the effectiveness of our test-time adaptation in cross-domain scenarios. The 3D-Front dataset has not been used in previous feed-forward methods [16, 22].



Figure S8: **Limitation of model dependencies.** The green square indicates object occlusion in the input image, which disrupts segmentation, leading to the disappearance of the reconstruction mesh.



Figure S9: **Limitation of transparency or reflection surface.** The surface texture of the input image is transparent and features special material properties of the mesh material. As a result, both the output geometry and texture are degraded.

4 In-depth Analysis of Limitation and Discussion

We conduct in-depth analyses of limitations and discussions that we could not discuss due to the length limitations of the main paper. Specifically, we present some failure cases of our method and discuss the future direction of improvement.

4.1 Failure Cases

Model dependencies. Our model utilizes the initial mesh and viewpoint predictions from the Image-to-3D module as the starting point for single-view image to 3D textured mesh reconstruction. It implies that our single-view to 3D capabilities are constrained by the capacity of the Image-to-3D module (*e.g.*, it only functions for categories where viewpoint prediction is feasible). Furthermore, we require images segmented to include only the object of interest to utilize the multi-view diffusion model. Therefore, the quality of segmentation directly impacts the quality of 3D reconstruction as shown in Fig. S8.

Transparency or reflection surface. Reconstructing 3D objects from single-view images has been a long-standing challenge. In addition, estimating PBR (Physically-Based Rendering) materials from single-view images presents an ill-posed problem, as there is inherent ambiguity between the diffuse component and lighting. In particular, the models currently in use assume microfacet surfaces [24]. Therefore, for instances with special material properties involving transparency or reflection, the texture optimization tends to degrade, resulting in sub-optimal geometry updates as shown in Fig. S9.

4.2 Discussion

We believe that expanding the Image-to-3D module into a more robust one capable of handling a larger class vocabulary could overcome model dependency issues despite the dependencies on the model in use. Because our test-time adaptation stage has the capability to category generalization as shown in Fig. S10. Additionally, address-

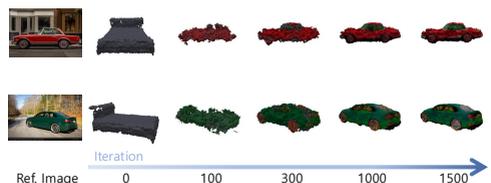


Figure S10: **Possibility of category extension.**

ing the degradation in reconstruction quality due to special reflection surfaces might be achievable through further exploration and application of complex material modeling and rendering equations in the future. Our research is practical in that it introduces a pipeline capable of operating in previously unseen out-of-distribution scenarios, especially in real-scene scenarios, which were not extensively considered in prior studies and can work for various viewpoint conditions in real images, which is different from existing generative prior methods [11, 14, 19]. We believe that our work can serve as a stepping stone for the advancement of single-view to 3D reconstruction methods that operate effectively in real scenarios.

References

- [1] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. *arXiv preprint arXiv:2303.13873*, 2023.
- [2] Yixin Chen, Junfeng Ni, Nan Jiang, Yaowei Zhang, Yixin Zhu, and Siyuan Huang. Single-view 3d scene reconstruction with high-fidelity shape and texture. *arXiv preprint arXiv:2311.00457*, 2023.
- [3] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. URL <http://www.blender.org>.
- [4] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023.
- [5] 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 *CVPR*, 2023.
- [6] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *ICCV*, 2021.
- [7] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. In *NeurIPS*, 2023.
- [8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.
- [9] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM TOG*, 39(6):1–14, 2020.
- [10] Haolin Liu, Yujian Zheng, Guanying Chen, Shuguang Cui, and Xiaoguang Han. Towards high-fidelity single-view holistic reconstruction of indoor scenes. In *ECCV*. Springer, 2022.

- [11] 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. *arXiv preprint arXiv:2303.11328*, 2023.
- [12] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.
- [13] Stephen McAuley, Stephen Hill, Naty Hoffman, Yoshiharu Gotanda, Brian Smits, Brent Burley, and Adam Martinez. Practical physically-based shading in film and game production. In *ACM SIGGRAPH 2012 Courses*, pages 1–7. 2012.
- [14] Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Realfusion: 360deg reconstruction of any object from a single image. In *CVPR*, 2023.
- [15] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *CVPR*, 2022.
- [16] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *CVPR*, 2020.
- [17] Poliigon. Poliigon. <https://www.poliigon.com/>.
- [18] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *CVPR*, 2018.
- [19] 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. *arXiv preprint arXiv:2303.14184*, 2023.
- [20] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, 2007.
- [21] Jonathan Young. xatlas. <https://github.com/jpcy/xatlas>, 2021.
- [22] Cheng Zhang, Zhaopeng Cui, Yinda Zhang, Bing Zeng, Marc Pollefeys, and Shuaicheng Liu. Holistic 3d scene understanding from a single image with implicit representation. In *CVPR*, 2021.
- [23] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018.