

SynthDrive: Scalable *Real2Sim2Real* Sensor Simulation Pipeline for High-Fidelity Asset Generation and Driving Data Synthesis

Zhengqing Chen^{1,*}, Ruohong Mei^{1,*}, Xiaoyang Guo^{1,†}, Qingjie Wang¹, Yubin Hu², Wei Yin¹, Weiqiang Ren¹ and Qian Zhang¹

Abstract—In the field of autonomous driving, sensor simulation is essential for generating rare and diverse scenarios that are difficult to capture in real-world environments. Current solutions fall into two categories: 1) CG-based methods, such as CARLA, which lack diversity and struggle to scale to the vast array of rare cases required for robust perception training; and 2) learning-based approaches, such as NeuSim, which are limited to specific object categories (vehicles) and require extensive multi-sensor data, hindering their applicability to generic objects. To address these limitations, we propose *SynthDrive*, a scalable “*real2sim2real*” system that leverages 3D generation to automate asset mining, generation, and rare-case data synthesis.

Our framework introduces two key innovations: 1) *Automated Rare-Case Mining and Synthesis*. Given a text prompt describing specific objects, *SynthDrive* automatically mines image data from the Internet and then generates corresponding high-fidelity 3D assets, which eliminates the need for costly manual data collection. By integrating these assets into existing street-view data, our pipeline produces photorealistic rare-case data, supporting rapid scaling to diverse assets including irregular obstacles and temporary traffic facilities. 2) *High-Fidelity 3D Generation*. We propose a hybrid asset generation pipeline that combines a geometry-aware LRM, iterative mesh optimization, and an improved texture fusion algorithm. Our approach achieves 0.0164 Chamfer Distance on the GSO dataset, outperforming InstantMesh by 14.1% in geometry accuracy, and achieves 19.05 PSNR (vs 16.84) for texture quality. This enables fine geometry details and high-resolution texture generation, which is essential for perception model training. Experiments demonstrate that *SynthDrive*-generated data improves the performance of downstream perception tasks (2D and 3D detection on rare objects) by 2-4% mAP. *SynthDrive* greatly lowers the data production cost and improves the diversity for corner-case data generation, showcasing extensive potential applications in the field of autonomous driving.

I. INTRODUCTION

The development of robust autonomous driving systems underscores the critical need for the coverage and diversity of training data and rare scenarios, particularly those involving safety-critical corner cases. However, relying solely on collecting data from the real world is insufficient in capturing all rare objects and corner cases [3]. Therefore, simulating these infrequent traffic scenarios is paramount for the development of robust and secure vehicles.

Existing computer graphics-based software, such as Carla [1], use graphic rendering engines to depict these corner cases. However, they heavily rely on manually crafted 3D assets to construct specific driving scenes. 3D assets in such systems typically encompass a limited range of

textures and lack realism, which restricts their capability to fully replicate the complexity and diversity of real-world environments. Another line of research works [2], [4]–[6] employ 3D reconstruction techniques instead of handcrafting to create 3D assets, which greatly improves the realism of 3D assets. The reconstructed 3D models are then inserted into existing driving scenes to create rare scenarios. To further improve realism, lighting estimation [7]–[9] and style transfer [10] can be additionally applied to ensure the correctness of environment lighting and realism. However, these methods investigate 3D reconstruction only for limited object categories like vehicles and require complex multi-sensor inputs, lacking flexibility and scalability to generic objects. Different from previous methods, recent image-to-3D methods [11]–[19] could generate diverse 3D assets from single input images, offering the potential to address the above issues in current data synthesis approaches.

Inspired by the above observations, we introduce *SynthDrive* (Fig. 2), a “*real2sim2real*” data generation pipeline to overcome the realism and scalability issues in existing pipelines [1], [2], [5], [6]. The core concept of this approach is *fully-automated* rare-data mining and synthesis. Our objectives address four critical gaps in existing methods: 1) **Generic Object Reconstruction**: Unlike NeuSim [2] and CADSim [5], which are confined to limited object categories, our system supports arbitrary objects (e.g., portable traffic lights, fallen debris) through automated asset mining. 2) **Low Data & Computation Cost**: In contrast to [7] requiring complex sensor inputs, *SynthDrive* reconstructs assets from a single image within 1 minute, without the need for additional manual data collection. 3) **High-Fidelity Geometry & Texture**: the proposed hybrid image-to-3D pipeline incorporates robust geometric and texture details, which is more suitable for perception model training compared with [14], [20].

To achieve these objectives, *SynthDrive* is built with three core modules: **Asset Mining, 3D Asset generation, and Scenario Synthesis**. Given a text prompt describing specific objects (e.g., “portable traffic lights” or “irregular obstacles”), our system automatically mines image data from existing driving data and web-collected sources using CLIP-guided [21] retrieval. This eliminates the need for manual data collection and ensures the coverage of diverse rare objects for simulation. The key of generic 3D asset reconstruction lies in our high-fidelity 3D asset generation module, which transforms single images into detailed 3D models. We propose a hybrid image-to-3D pipeline that combines a geometry-aware LRM, iterative mesh optimization, and an

¹ Horizon Robotics. ² Tsinghua University

* Equal contribution † Corresponding Author.



Fig. 1. Comparison of different approaches for autonomous driving simulation. Left: CARLA [1], a CG-based platform with manually crafted assets, lacking realism and scalability. Middle: NeuSim [2], a vehicle-only asset reconstruction approach requiring complex multi-modality images and LiDAR data. Right: Our approach, a fully automated data synthesis framework capable of generating diverse generic assets. The figure shows the automatically-constructed asset bank that includes traffic elements, vehicles, and road infrastructure.

improved texture fusion algorithm. Specifically, starting with a single input image, we first extract high-quality texture and geometry cues using a multi-view diffusion model [16] and an image-to-normal diffusion model [22]. The multi-view texture cues are utilized to initialize a coarse 3D model with a feed-forward large reconstruction model [14], and the geometric surface normal cues are utilized to refine the coarse model through a normal-supervised explicit mesh optimization process, enabled by differentiable surface normal rendering. To further improve texture quality, we up-sample the texture cues by diffusion-based image super-resolution [23]. The high-resolution images are then comprehensively mapped and fused across multiple views to produce the final textured 3D models, maintaining high-fidelity details.

Finally, the generated 3D assets are integrated into existing driving videos or reconstructed simulation environments for further synthetic data generation and closed-loop simulations. A complex HDR lighting estimation and post-video harmonizer are employed to ensure the correctness of shadows and colors. The road surface geometry is also reconstructed to ensure the correct placement of inserted 3D assets. It offers enhanced realism and diversity for the simulated scenarios, which are crucial for the rapid development of autonomous driving systems. To sum up, our contributions are three-fold:

- We introduce *SynthDrive*, a cost-effective and scalable “*real2sim2real*” pipeline for automatic asset library construction and data synthesis for autonomous driving.
- We propose a novel single-view reconstruction approach with enhanced mesh geometry optimization and detailed texture fusion to create high-fidelity and high-resolution generic 3D assets.
- We conduct extensive experiments on public datasets. Results demonstrate that our 3D asset generation approach surpasses state-of-the-art approaches in terms of geometric accuracy, rendering quality, view generalization, and versatility. Our synthesis data significantly improves the performance of downstream perception tasks (2D and 3D detection on rare objects).

II. RELATED WORK

A. Asset Reconstruction

In the domain of autonomous driving simulations, various methodologies have been explored for 3D asset reconstruction and generation. As illustrated in Fig. 1, Computer

Graphics (CG)-based platforms like CARLA [1] offer comprehensive simulation tools but rely heavily on manually crafted 3D assets, which limits its scalability and adaptability. Techniques based on neural radiance fields (NeRFs) [24], [27]–[30], NeuS [25], [31]–[34] and recent 3D Gaussian Splatting [26], [35]–[37] require high-quality 360-view scans of objects. However, objects in autonomous driving data are usually sparsely observed, leading to suboptimal geometry and texture quality in occluded areas.

To address the above issues, several approaches [5], [6] incorporate additional priors to enhance vehicle reconstruction quality. GeoSim [6] incorporates multi-view supervisions and LiDAR-depth observations into a mesh reconstruction network for 3D asset generation. CADSim [5] combines part-aware CAD models with differentiable rendering to reconstruct vehicle geometry. NeuSim [2] utilizes a neural volume rendering approach to reconstruct vehicles from in-the-wild images and LiDAR data. However, these methods primarily focus on the reconstruction of vehicle assets, relying on extensive multi-view and multi-sensor data, which limits their applicability to specific objects like cars. In contrast, our method requires only a single image to reconstruct the object with significantly finer details. Concurrently, research works like 3DRealCar [38] and DreamCar [39] collect 360-degree car datasets and employ diffusion priors for 3D asset generation. However, their focus remains solely on vehicles, thus limiting their broader applications.

To synthesize corner-case data, the reconstructed 3D assets are inserted into existing driving scenes for scene editing. To enhance realism, LightSim [9] proposes to synthesize data under different lightings. [8] recovers a 5D HDR light field for relighting inserted objects. FEGR [7] employs an inverse rendering technique to recover the lighting conditions of existing images, enhancing the virtual object insertion realism. However, these methods did not address the scalability issues for asset library construction.

B. Single View 3D Reconstruction

Single view 3D reconstruction [11] aims to reconstruct objects from only a single image as input. One line of work [11], [40]–[42] utilizes diffusion generative priors from pretrained image diffusion models [43], [44], guided by score distillation sampling loss (SDS) [11]. However, these approaches usually suffer from “multi-face problems” and time-consuming per-scene optimization. To accelerate the generation, Zero123 [15] demonstrates that diffusion

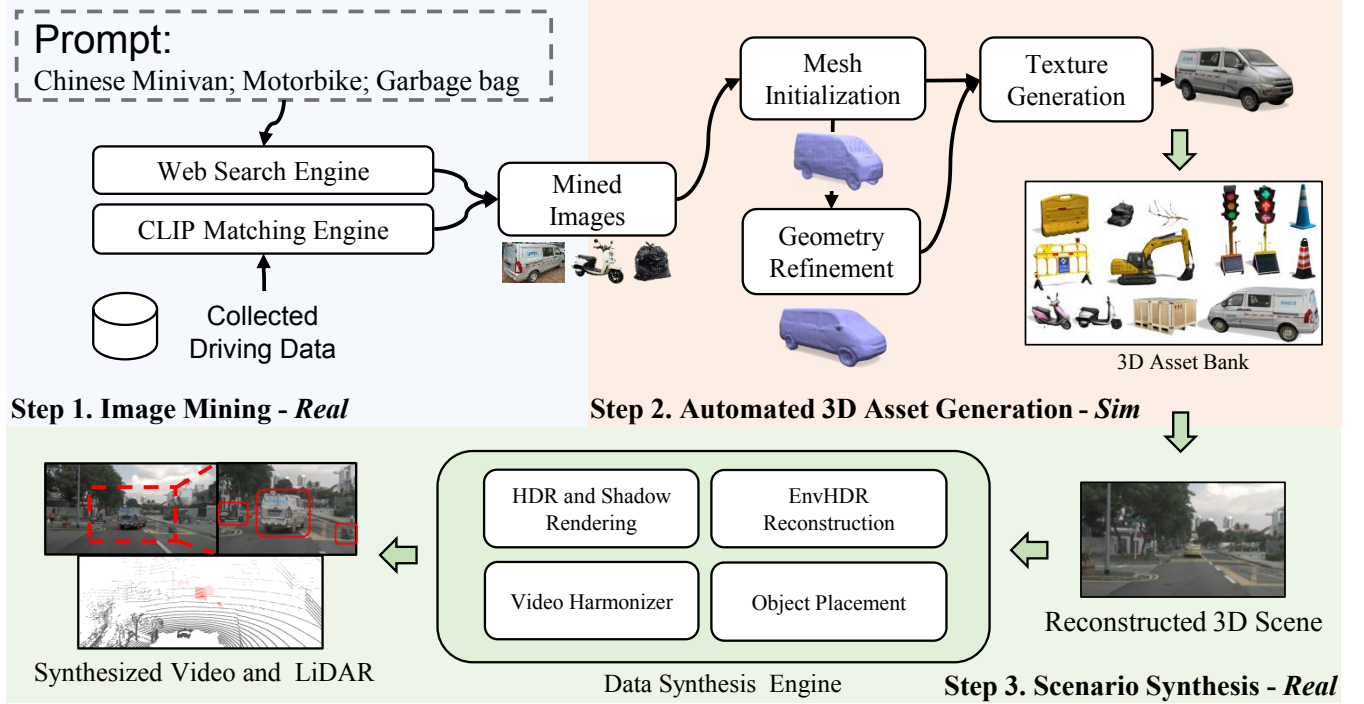


Fig. 2. The Framework of *SynthDrive*. 1) Given a text *prompt*, our system automatically mines asset images from existing driving data via CLIP [21], or from web search engines. 2) High-quality 3D assets are reconstructed from these images using the proposed hybrid multi-stage image-to-3D algorithm, producing a comprehensive 3D asset bank. 3) Synthetic data is finally rendered by blending real images (or reconstructed 3D scenes [24]–[26]) with 3D assets queried from the asset bank in a large-scale scalable way.

models [43] can be fine-tuned to generate novel view images from a single-view input conditioned on target camera poses. Subsequent works [16], [45], [46] further improve the consistency of multi-view predictions by simultaneously generating multi-views of an object. 3D inconsistency from those generated multi-view images can impact the geometric quality of the 3D models. Wonder3D [20] proposes to predict multi-view normals as supplementary signals and adapt SDF-based 3D reconstruction techniques to ensure robustness. One2345 [17], [18] directly regresses plausible 3D shapes from the images generated from Zero123 [15].

More recently, large reconstruction models [12]–[14], [47], [48] have demonstrated that 3D generation is feasible using feed-forward networks. Large transformer models [49] are trained on large-scale 3D datasets [50], [51] to predict diverse 3D representations, such as NeRF [47], 3D Gaussian [13], and mesh [14], [48]. Although the above methods achieve significant efficiency, the geometric and texture quality remains suboptimal for asset generation in self-driving simulations. It is crucial to ensure asset details to bridge domain gaps, therefore, in this paper, we extensively explore to improve the geometric and textural details in our proposed framework.

III. METHODOLOGY

We propose *SynthDrive*, an automated data generation framework tailored for rare data synthesis and autonomous driving simulation. As illustrated in Fig. 2, the framework consists of three core modules: **image mining**, **automated 3D asset generation**, and **multi-modal sensor simulations**. The framework begins with CLIP-guided retrieval to

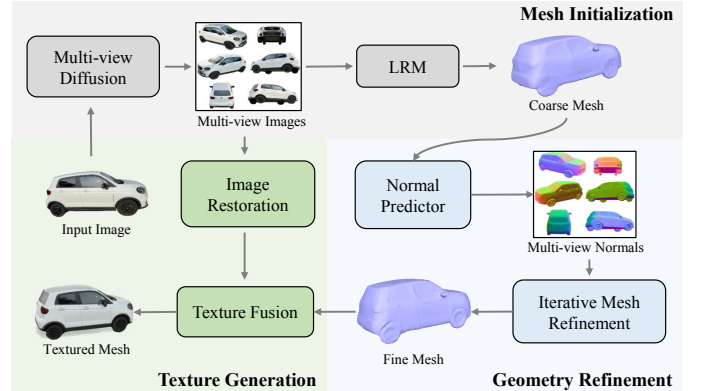


Fig. 3. Illustration of the Proposed 3D Asset Generation Pipeline. Given a single input image, multi-view images and normal maps are generated using a multi-view diffusion model [16] and a normal estimation model [22]. Starting with a coarse mesh reconstructed from LRM, the geometry is refined by iterative normal-aware mesh optimization. The final texture is enhanced with an improved texture fusion strategy to preserve high-resolution details.

mine diverse image data from multiple sources (Sec. III-A). Next, a hybrid image-to-3D pipeline combines iterative mesh refinement and advanced texturing mapping to generate high-fidelity 3D assets from single-view images (Sec. III-B). Finally, the assets are integrated into driving scenes through relighting and video harmonization, combined with 3D reconstruction for accurate object placement (Sec. III-C).

A. Image Mining

The pipeline begins with a sophisticated image-mining process. This phase is crucial for reconstructing rare object categories that are essential for robust autonomous driving

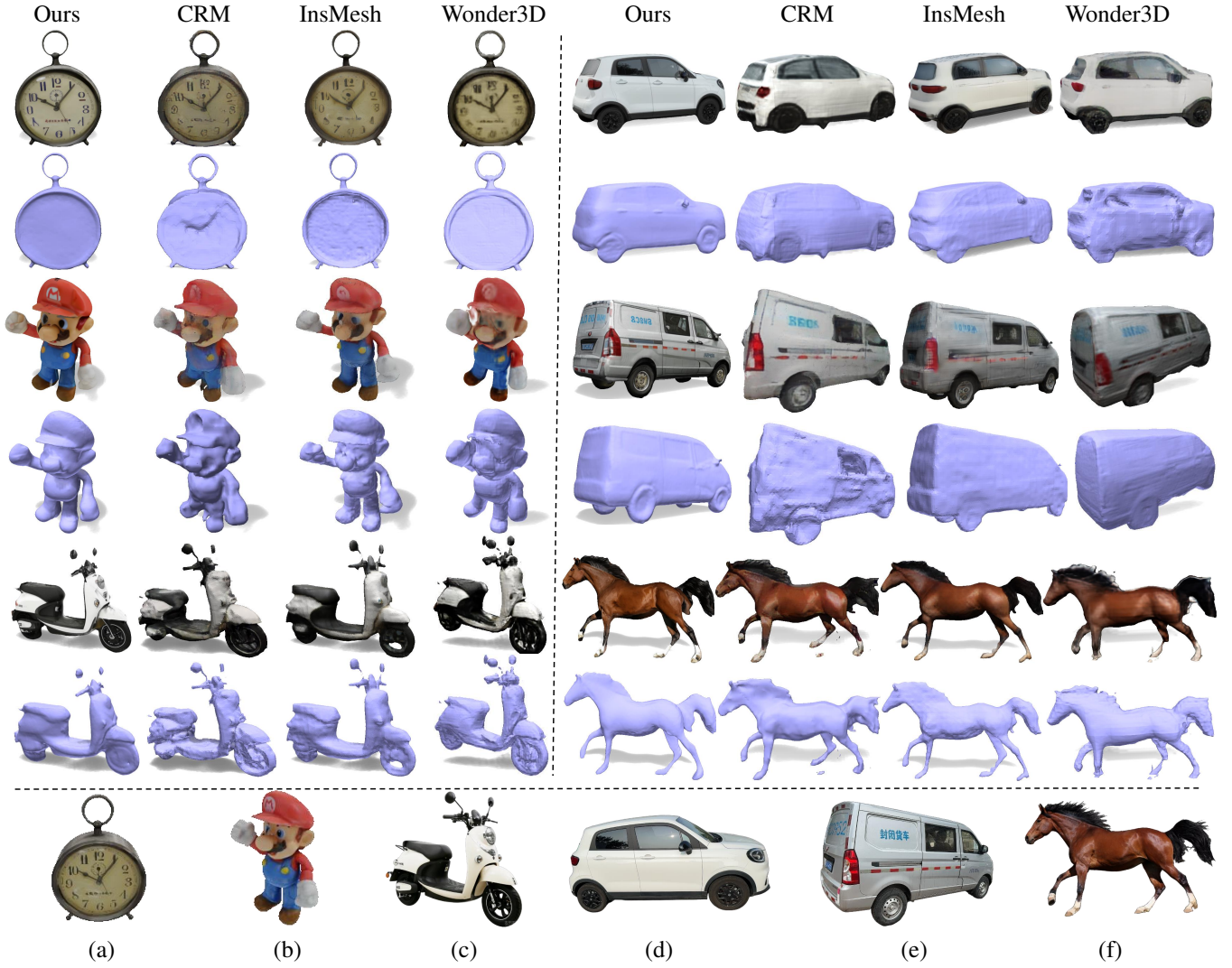


Fig. 4. Qualitative Comparison of Different Methods. (a)-(f) are inputs. Our mesh shows superior geometry and texture quality. Zoom in for more details.

systems. Given existing driving data, CLIP embeddings [21] of images are pre-computed to facilitate *text-to-image* retrieval. Text *prompts* (e.g., “portable traffic lights” and “irregular obstacles”) are used to guide the retrieval of images for rare objects. Additionally, images are sourced from web search engines to ensure broader diversity and coverage of image results, complementing the existing database with a wider complexity and variety of challenging scenarios.

B. Automatic 3D Asset Generation from Mined Images

Mesh Initialization. Generating multi-view images from a single image is fundamental to single-image 3D reconstruction. We employ Zero123++ [16] for multi-view image generation due to its ability to model multiple fixed viewpoints simultaneously, enhancing multi-view consistency and yielding high-quality multi-view images. These images are then used to initialize a coarse 3D model via InstantMesh [14], a transformer-based large reconstruction model (LRM). While LRM networks [13], [14], [47] generally produce meshes with good topology, they are limited by spatial resolution and lack of geometry details, leading to suboptimal reconstruction precision (see Fig. 3 and Fig. 4). To address this, we use

StableNormal [22] to predict normal maps from multi-view images as geometric cues for subsequent mesh optimization.

Iterative Mesh Refinement. Inspired by prior works [20], [52], [53], we apply a mesh-based optimization that refines the initial coarse mesh using differentiable rendering. Given the initial coarse mesh, it first undergoes differentiable rendering [54] to generate normal maps and silhouette maps, which are used to compute loss with multi-view observations. The vertex positions are updated based on the gradients, followed by explicit surface remeshing, which iteratively applies edge-split, edge-collapse, smoothing, and relocating [53]. During each iteration, the initial mesh is subdivided, optimized, and smoothed, enabling the expression of finer geometric details and steering optimization away from local minima. The loss function of iterative mesh refinement consists of three components:

$$L = L_n + \lambda_{mask} L_{mask} + \lambda_{lap} L_{lap},$$

where $L_n = \sum_i \left\| \hat{N}_i - N_i^{pred} \right\|_2^2$ is the L2 loss of normal maps from different views, $L_{mask} = \sum_i \left\| \hat{M}_i - M_i^{pred} \right\|_2^2$ represents

the alpha mask loss between the rendered images and the silhouette maps. Additionally, we introduce a Laplacian smoothness [55] term L_{lap} to constrain the consistency of surface normals, resulting in a smoother mesh.

Texture Generation. Realistic appearance is crucial for autonomous driving simulation assets, as it enhances the realism of simulation scenes and reduces the domain gap between synthetic and real data. To address the resolution limitations of multi-view images generated by diffusion models, we adopt a diffusion-based image super-resolution model [23] to upsample the images by a factor of two to recover more texture details.

Unlike traditional methods that rely on the initial colors provided by LRM networks, we develop an explicit texture fusion algorithm to faithfully preserve texture details from the upsampled multi-view images. Specifically, each mesh vertex is initially assigned a color from the view with maximum cosine similarity between the vertex normal and the viewing direction. This strategy ensures optimal texture quality by prioritizing views that provide the most direct observation of each surface point. Vertices at the seams between viewpoints are smoothed using Laplacian smoothing [55] to reduce color inconsistencies. Finally, we fill the colors of invisible vertices by propagating colors from visible vertices to adjacent invisible ones, producing a completely colored mesh with high-resolution details.

C. Scenario Synthesis

The refined 3D assets are integrated into driving scenes through a multi-stage pipeline: **1) 3D Reconstruction and Object Placement.** The road surface is reconstructed from multi-frame aggregated LiDAR scans using [56]. If LiDAR data is not available, a learning-based multi-view stereo [57] is employed for 3D geometry reconstruction. To prevent collisions between inserted 3D assets and existing road users, an object tracker estimates the trajectories of vehicles and pedestrians. Assets are placed only in collision-free regions. **2) HDR Environment Lighting and Foreground Rendering.** Foreground assets are rendered using a ray-casting rendering engine [58]. To ensure accurate rendering of lighting and shadows, High dynamic range (HDR) environment lighting is composed by blending 6 surrounding LDR images with direct sunlight. Depth estimation and semantic segmentation are applied to construct occlusion masks between foreground assets and static backgrounds, ensuring realistic layering. **3) Image Composition and Post-processing.** The rendered foreground assets are then composited with the background image or 3D models [25], [26] through alpha blending. To further enhance realism, a video-based harmonizer [59] is employed to adjust the color temperature, contrast, and saturation of foreground renderings, resulting in more harmonious composition results. Visualization examples are presented in Fig. 2 and Fig. 5.

IV. EXPERIMENTS

A. Experimental Settings

Dataset. We evaluated the performance of our proposed hybrid 3D asset generation method using two public datasets:

TABLE I
RESULTS ON GSO DATASET. **BOLD:** BEST. CD: CHAMFER DISTANCE.

Method	CD↓	Volume IoU↑	PSNR↑	SSIM↑	LPIPS↓
Magic123	0.0516	0.4528	12.69	0.7984	0.2442
SyncDreamer	0.0261	0.5421	14.00	0.8165	0.2591
Wonder3D	0.0199	0.6244	13.31	0.8121	0.2554
CRM	0.0173	0.6286	16.22	0.8381	0.2143
InsMesh	0.0191	0.5810	16.84	0.8408	0.1749
Ours	0.0164	0.6731	19.05	0.8724	0.1255

TABLE II
RESULTS FOR DIFFERENT METHODS ON REALCAR360 **NORMAL/HARD** EVALUATION SET. **BOLD:** BEST. CD: CHAMFER DISTANCE.

Method	CD↓	Volume IoU↑	PSNR↑	SSIM↑	LPIPS↓
Wonder3D	0.0283/0.0322	0.48/0.40	14.50/13.90	0.768/0.793	0.204/0.263
CRM	0.0240/0.0330	0.54/0.43	15.29/14.42	0.808/0.775	0.167/0.180
InsMesh	0.0190/0.0256	0.69/0.56	16.53/14.76	0.845/0.794	0.157/0.177
Ours	0.0170/0.0184	0.76/0.70	16.98/16.51	0.857/0.851	0.128/0.133

Google Scanned Objects (GSO) [60] and 3DRealCar [38]. The GSO dataset contains approximately 1,000 scanned objects. Following the valuation protocol of SyncDreamer [45], 30 selected objects from various categories were chosen as the evaluation dataset. For the 3DRealCar dataset, which contains real-world car captures, 15 vehicles with diverse appearances and shapes were selected. For each car, we chose one normal viewpoint and one difficult viewpoint to evaluate the texture quality of reconstructed meshes, resulting in the *normal* and the *hard* evaluation set.

We also evaluated the framework’s effectiveness on two downstream tasks. For 2D detection, we use our self-collected real-driving data to evaluate performance on rare categories, specifically the Traffic A-frame sign and the warning tripod. The training set consists of 160,000 and 15,000 2D bounding box annotations for these categories, respectively, while the test set contains 2,400 and 2,100 annotated samples. For 3D detection, we employed the NuScenes dataset, focusing on three rare categories (traffic cones, construction vehicles, and motorcycles). We use 700 street-view scenes for training and 150 for validation.

Baselines. For 3D asset generation, we compare our method with the following baselines, 1) Magic123 [40]: A SDS-based 2D lifting method for single-view reconstruction. 2) SyncDreamer [45]: A synchronized multi-view diffusion model for 3D reconstruction. 3) Wonder3D [20]: A multi-view cross-domain 2D diffusion model. 4) CRM [48]: A convolutional reconstruction model that generates multi-view images and canonical coordinate maps (CCMs). 5) InstantMesh [14]: A feed-forward framework trained with differentiable mesh rendering for high-quality 3D mesh generation. For 2D detection, we used FCOS [61] as our baseline, and for 3D detection, BEVFormer [62] was adopted.

Implementation Details. For 3D asset generation, the number of mesh refinement steps is set to 20, with hyperparameters for different losses set as $\lambda_{mask} = 1$ and $\lambda_{lap} = 0.5$. The inference time for the asset generation method is around 30 seconds on a single RTX 3090 GPU. For data synthesis, it

takes approximately 20 minutes to synthesize a 2-min video (12 FPS) on a single RTX 3090 GPU.

B. Evaluation of 3D Asset Generation

Qualitative Results. We selected representative cases from GSO, 3DRealCar, and web image datasets for comparison, as shown in Figure 4. Our method demonstrates superior geometry quality and texture fidelity compared to baseline approaches. For example, our reconstructions capture clear clock numerals in Fig. 4(a) and precise vehicle details like rear-view mirrors and wheel contours in Fig. 4(d)(e). Even with internet-sourced data exhibiting varied lighting and geometric complexities, our method maintains high-quality reconstruction, demonstrating excellent viewpoint invariance and generalization capability. In contrast, other methods suffer from noisy geometry (CRM), lack of fine details (InstantMesh), or limited viewpoint generalization (Wonder3D). These improvements stem directly from our iterative mesh optimization and enhanced texture fusion strategy.

Quantitative Results. On the GSO dataset, our method outperforms all baseline approaches across geometric and visual quality metrics. Compared to the second-best method, we achieve a 14.1% decrease in Chamfer Distance and a 7.1% improvement in Volume IoU. For visual quality, our approach yields significant gains with a 13.1% increase in PSNR, 3.8% improvement in SSIM, and 28.2% reduction in LPIPS. On the RealCar360 dataset’s normal/difficult viewpoint evaluation, our method not only achieves the best results on both subsets but also exhibits minimal performance degradation on difficult viewpoints, validating its robustness in 360-degree view generalization.

TABLE III

COMPARISON OF 2D DETECTION ON THE SELF-COLLECTED DATASET.

Training data	A-frame Sign		Warning Tripod	
	AP↑(%)	AR↑ (%)	AP↑ (%)	AR↑ (%)
real	24.3	39.2	72.4	73.3
real+syn data	27.4(+3.1)	43.1(+3.9)	76.5(+4.1)	77.1(+3.8)

TABLE IV

COMPARISON OF 3D DETECTION ON NuSCENES DATASET.

Training data	3D Detection		Per-category AP↑		
	NDS↑	mAP↑	motorcycle	cone	vehicle
real	40.7	33.5	36.1	54.1	10.5
real+20%syn data	42.4 (+1.7)	35.6 (+2.1)	40.1 (+4.0)	54.2 (+0.1)	12.7 (+2.2)
real+50%syn data	41.6 (+0.9)	35.2 (+1.7)	40.5 (+4.4)	54.7 (+0.6)	10.4 (-0.1)

C. Downstream Evaluation on 2D and 3D Object Detection

3D Object Detection: To evaluate the effectiveness of our synthetic data for rare object detection, we generated additional training samples and mixed them with real NuScenes data in varying proportions to train BEVFormer [62]. As shown in Table IV, incorporating 20% additional synthetic data significantly improved performance, with NDS increasing by 1.7% and mAP by 2.1%. The improvement was particularly pronounced for motorcycles (+4.0%), which have limited training samples in real datasets. These results demonstrate the effectiveness of synthesized data for rare object detection with appropriate mixing proportions.

2D Object Detection: To further evaluate the effectiveness of our synthesized data on the 2D detection task, we augmented a self-collected driving dataset by synthesizing 40,026 A-frame signs and 19,544 warning tripods, along with the corresponding 2D automated annotations. The synthesized data accounted for approximately 20% of the total real training data. As shown in Table III, adding synthetic data gives superior performance on rare categories, with a 3.5% increase in average precision (AP) and a 4.0% increase in average recall (AR).

These results demonstrate that our high-fidelity synthetic data effectively enhances both 2D and 3D detection performance, particularly for rare and safety-critical categories in autonomous driving scenarios.

D. Ablation Study

1) *Performance w.r.t Iteration Steps of Mesh Refinement:* As shown in Fig. 6, we report the F-score and Chamfer Distance at each iteration to measure the convergence of mesh refinement. It can be observed that the geometric metrics converge after around 20 iterations. With an increasing number of iterations, the number of mesh faces increases, which raises the difficulty of geometry optimization. Therefore, to balance accuracy and efficiency, we perform 20 steps for each mesh.

2) *Effect of Texture Fusion:* To demonstrate the importance of texture fusion strategy, we render images of the same mesh with and without texture fusion. As shown in Fig. 7, the mesh with texture fusion appears clear and has sharp edges, while the mesh without texture fusion exhibits blurriness. The high resolution of the texture ensures the clearness and fidelity of synthesized data.

E. Asset Bank and Scenario Synthesis

Through our automated asset mining and generation framework, we have constructed an Asset Bank containing over 2000 high-quality 3D assets. This asset bank encompasses various non-whitelist objects encountered in autonomous driving scenarios, including road obstacles, temporary traffic facilities (portable traffic lights, movable signage), irregular vehicles, and large and small fallen objects. These assets are crucial for testing the perception robustness and capabilities of autonomous driving systems in non-typical and safety-critical scenarios.

Figure 5 shows the results of scenario synthesis on the nuScenes dataset using our generated 3D assets. The left side presents the camera simulation results, while the right side shows the corresponding LiDAR simulation results. In Scene A, we inserted oncoming vehicles, traffic cones, temporary traffic lights, and obstacle boxes to simulate the generalizability of autonomous driving vehicles to temporary changes in the environment. In Scene B, we introduced water barriers, barricades, and bulldozers to test the perception capabilities of autonomous driving systems for irregular vehicles and obstacles. Our approach demonstrates the ability to perform realistic sensor simulation for autonomous driving

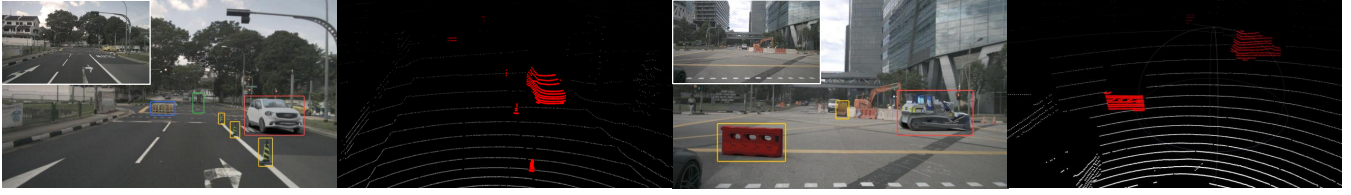


Fig. 5. Synthetic Examples with Camera and LiDAR Simulation. High-quality mined 3D assets enable straightforward synthesis of rare driving scenarios.

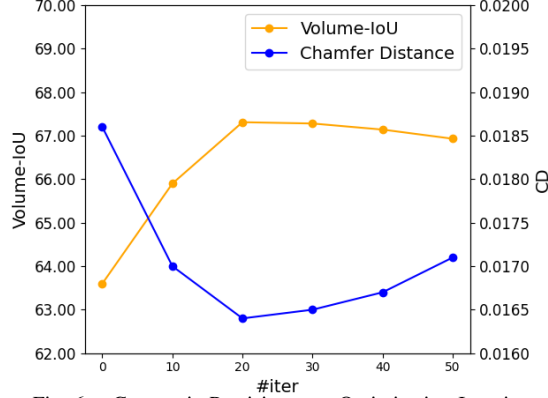


Fig. 6. Geometric Precision w.r.t Optimization Iteration.

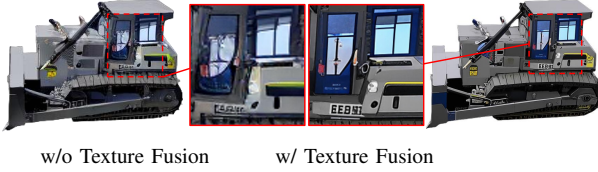


Fig. 7. Effect of the Proposed Texture Fusion Strategy

across various scenarios, particularly those involving safety-critical situations.

V. CONCLUSION

In this work, we introduced a novel framework for automated asset mining and synthetic data generation designed for autonomous driving simulations. Our methodology leverages state-of-the-art 3D generation techniques, alongside improved geometry refinement and texture fusion strategies. Our pipeline enables automatic asset mining and 3D asset reconstruction from both data collected from real-world and web search engines, providing high-quality, scalable, and low-cost digital assets. Experiment results demonstrated superior scalability, robustness, and geometry accuracy over existing technologies. We hope this work will inspire the self-driving community, contributing to safer and more robust autonomous driving perception.

REFERENCES

- [1] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: An open urban driving simulator,” in *Conference on robot learning*. PMLR, 2017, pp. 1–16.
- [2] Z. Yang, S. Manivasagam, Y. Chen, J. Wang, R. Hu, and R. Urtasun, “Reconstructing objects in-the-wild for realistic sensor simulation,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 661–11 668.
- [3] J. Wang, A. Pun, J. Tu, S. Manivasagam, A. Sadat, S. Casas, M. Ren, and R. Urtasun, “Advsim: Generating safety-critical scenarios for self-driving vehicles,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9909–9918.
- [4] J. Zhang, G. Yang, S. Tulsiani, and D. Ramanan, “Ners: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 835–29 847, 2021.
- [5] J. Wang, S. Manivasagam, Y. Chen, Z. Yang, I. A. B arsan, A. J. Yang, W.-C. Ma, and R. Urtasun, “Cadsim: Robust and scalable in-the-wild 3d reconstruction for controllable sensor simulation,” *arXiv preprint arXiv:2311.01447*, 2023.
- [6] Y. Chen, F. Rong, S. Duggal, S. Wang, X. Yan, S. Manivasagam, S. Xue, E. Yumer, and R. Urtasun, “Geosim: Realistic video simulation via geometry-aware composition for self-driving,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 7230–7240.
- [7] Z. Wang, T. Shen, J. Gao, S. Huang, J. Munkberg, J. Hasselgren, Z. Gojcic, W. Chen, and S. Fidler, “Neural fields meet explicit geometric representations for inverse rendering of urban scenes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8370–8380.
- [8] Z. Wang, W. Chen, D. Acuna, J. Kautz, and S. Fidler, “Neural light field estimation for street scenes with differentiable virtual object insertion,” in *European Conference on Computer Vision*. Springer, 2022, pp. 380–397.
- [9] A. Pun, G. Sun, J. Wang, Y. Chen, Z. Yang, S. Manivasagam, W.-C. Ma, and R. Urtasun, “Lightsim: Neural lighting simulation for urban scenes,” *arXiv preprint arXiv:2312.06654*, 2023.
- [10] C. Bai, Z. Shao, G. Zhang, D. Liang, J. Yang, Z. Zhang, Y. Guo, C. Zhong, Y. Qiu, Z. Wang *et al.*, “Anything in any scene: Photorealistic video object insertion,” *arXiv preprint arXiv:2401.17509*, 2024.
- [11] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, “Dreamfusion: Text-to-3d using 2d diffusion,” *arXiv preprint arXiv:2209.14988*, 2022.
- [12] D. Tochilkin, D. Pankratz, Z. Liu, Z. Huang, A. Letts, Y. Li, D. Liang, C. Laforte, V. Jampani, and Y.-P. Cao, “Tripotr: Fast 3d object reconstruction from a single image,” *arXiv preprint arXiv:2403.02151*, 2024.
- [13] Y. Xu, Z. Shi, W. Yifan, H. Chen, C. Yang, S. Peng, Y. Shen, and G. Wetzstein, “Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation,” *arXiv preprint arXiv:2403.14621*, 2024.
- [14] J. Xu, W. Cheng, Y. Gao, X. Wang, S. Gao, and Y. Shan, “Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models,” *arXiv preprint arXiv:2404.07191*, 2024.
- [15] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, “Zero-1-to-3: Zero-shot one image to 3d object,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 9298–9309.
- [16] R. Shi, H. Chen, Z. Zhang, M. Liu, C. Xu, X. Wei, L. Chen, C. Zeng, and H. Su, “Zero123++: a single image to consistent multi-view diffusion base model,” 2023.
- [17] M. Liu, C. Xu, H. Jin, L. Chen, M. Varma T, Z. Xu, and H. Su, “One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [18] M. Liu, R. Shi, L. Chen, Z. Zhang, C. Xu, X. Wei, H. Chen, C. Zeng, J. Gu, and H. 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*, 2024, pp. 10 072–10 083.
- [19] K. Wu, F. Liu, Z. Cai, R. Yan, H. Wang, Y. Hu, Y. Duan, and K. Ma, “Unique3d: High-quality and efficient 3d mesh generation from a single image,” *arXiv preprint arXiv:2405.20343*, 2024.
- [20] X. Long, Y.-C. Guo, C. Lin, Y. Liu, Z. Dou, L. Liu, Y. Ma, S.-H. Zhang, M. Habermann, C. Theobalt *et al.*, “Wonder3d: Single image to 3d using cross-domain diffusion,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9970–9980.

- [21] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.
- [22] C. Ye, L. Qiu, X. Gu, Q. Zuo, Y. Wu, Z. Dong, L. Bo, Y. Xiu, and X. Han, “Stablenormal: Reducing diffusion variance for stable and sharp normal,” *arXiv preprint arXiv:2406.16864*, 2024.
- [23] V. Arkhipkin, A. Filatov, V. Vasilev, A. Maltseva, S. Azizov, I. Pavlov, J. Agafonova, A. Kuznetsov, and D. Dimitrov, “Kandinsky 3.0 technical report,” *arXiv preprint arXiv:2312.03511*, 2023.
- [24] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [25] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, “Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction,” *arXiv preprint arXiv:2106.10689*, 2021.
- [26] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering,” *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.
- [27] T. Müller, A. Evans, C. Schied, and A. Keller, “Instant neural graphics primitives with a multiresolution hash encoding,” *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.
- [28] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretschmar, “Block-nerf: Scalable large scene neural view synthesis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8248–8258.
- [29] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, “Mip-nerf 360: Unbounded anti-aliased neural radiance fields,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5470–5479.
- [30] —, “Zip-nerf: Anti-aliased grid-based neural radiance fields,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 697–19 705.
- [31] Z. Li, T. Müller, A. Evans, R. H. Taylor, M. Unberath, M.-Y. Liu, and C.-H. Lin, “Neuralangelo: High-fidelity neural surface reconstruction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8456–8465.
- [32] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger, “Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction,” *Advances in neural information processing systems*, vol. 35, pp. 25 018–25 032, 2022.
- [33] Z. Yang, Y. Chen, J. Wang, S. Manivasagam, W.-C. Ma, A. J. Yang, and R. Urtaun, “Unisim: A neural closed-loop sensor simulator,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1389–1399.
- [34] L. Yariv, P. Hedman, C. Reiser, D. Verbin, P. P. Srinivasan, R. Szeliski, J. T. Barron, and B. Mildenhall, “Baked sdf: Meshing neural sdf for real-time view synthesis,” in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023, pp. 1–9.
- [35] Z. Yu, A. Chen, B. Huang, T. Sattler, and A. Geiger, “Mip-splatting: Alias-free 3d gaussian splatting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 447–19 456.
- [36] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang, “Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 634–21 643.
- [37] Y. Yan, H. Lin, C. Zhou, W. Wang, H. Sun, K. Zhan, X. Lang, X. Zhou, and S. Peng, “Street gaussians for modeling dynamic urban scenes,” *arXiv preprint arXiv:2401.01339*, 2024.
- [38] X. Du, H. Sun, S. Wang, Z. Wu, H. Sheng, J. Ying, M. Lu, T. Zhu, K. Zhan, and X. Yu, “3drealcar: An in-the-wild rgb-d car dataset with 360-degree views,” *arXiv preprint arXiv:2406.04875*, 2024.
- [39] X. Du, H. Sun, M. Lu, T. Zhu, and X. Yu, “Dreamcar: Leveraging car-specific prior for in-the-wild 3d car reconstruction,” *arXiv preprint arXiv:2407.16988*, 2024.
- [40] G. Qian, J. Mai, A. Hamdi, J. Ren, A. Siarohin, B. Li, H.-Y. Lee, I. Skorokhodov, P. Wonka, S. 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.
- [41] J. Tang, T. Wang, B. Zhang, T. Zhang, R. Yi, L. Ma, and D. 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*, 2023, pp. 22 819–22 829.
- [42] H. Wang, X. Du, J. Li, R. A. Yeh, and G. Shakhnarovich, “Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 619–12 629.
- [43] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.
- [44] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach, “Sdxl: Improving latent diffusion models for high-resolution image synthesis,” *arXiv preprint arXiv:2307.01952*, 2023.
- [45] Y. Liu, C. Lin, Z. Zeng, X. Long, L. Liu, T. Komura, and W. Wang, “Syncdreamer: Generating multiview-consistent images from a single-view image,” *arXiv preprint arXiv:2309.03453*, 2023.
- [46] Y. Shi, P. Wang, J. Ye, M. Long, K. Li, and X. Yang, “Mvdream: Multiview diffusion for 3d generation,” *arXiv preprint arXiv:2308.16512*, 2023.
- [47] Y. Hong, K. Zhang, J. Gu, S. Bi, Y. Zhou, D. Liu, F. Liu, K. Sunkavalli, T. Bui, and H. Tan, “Lrm: Large reconstruction model for single image to 3d,” *arXiv preprint arXiv:2311.04400*, 2023.
- [48] Z. Wang, Y. Wang, Y. Chen, C. Xiang, S. Chen, D. Yu, C. Li, H. Su, and J. Zhu, “Crn: Single image to 3d textured mesh with convolutional reconstruction model,” *arXiv preprint arXiv:2403.05034*, 2024.
- [49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need (nips), 2017,” *arXiv preprint arXiv:1706.03762*, vol. 10, p. S0140525X16001837, 2017.
- [50] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. Vander-Bilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, “Objaverse: A universe of annotated 3d objects,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 142–13 153.
- [51] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre *et al.*, “Objaverse-xl: A universe of 10m+ 3d objects,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [52] Y. Lu, J. Zhang, S. Li, T. Fang, D. McKinnon, Y. Tsin, L. Quan, X. Cao, and Y. Yao, “Direct2. 5: Diverse text-to-3d generation via multi-view 2.5 d diffusion,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 8744–8753.
- [53] W. Pfaffinger, “Continuous remeshing for inverse rendering,” *Computer Animation and Virtual Worlds*, vol. 33, no. 5, p. e2101, 2022.
- [54] S. Laine, J. Hellsten, T. Karras, Y. Seol, J. Lehtinen, and T. Aila, “Modular primitives for high-performance differentiable rendering,” *ACM Transactions on Graphics (ToG)*, vol. 39, no. 6, pp. 1–14, 2020.
- [55] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, “Laplacian surface editing,” in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 2004, pp. 175–184.
- [56] J. Huang, Z. Gojcic, M. Atzmon, O. Litany, S. Fidler, and F. Williams, “Neural kernel surface reconstruction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4369–4379.
- [57] J. Cheng, W. Yin, K. Wang, X. Chen, S. Wang, and X. Yang, “Adaptive fusion of single-view and multi-view depth for autonomous driving,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 138–10 147.
- [58] “Blender,” <https://www.blender.org/>, 2024, accessed: yyyy-mm-dd.
- [59] Z. Ke, C. Sun, L. Zhu, K. Xu, and R. W. Lau, “Harmonizer: Learning to perform white-box image and video harmonization,” in *European Conference on Computer Vision*. Springer, 2022, pp. 690–706.
- [60] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke, “Google scanned objects: A high-quality dataset of 3d scanned household items,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2553–2560.
- [61] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: A simple and strong anchor-free object detector,” 2021.
- [62] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, “Bev-former: learning bird’s-eye-view representation from lidar-camera via spatiotemporal transformers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.