Learning in-the-wild 3D Vehicle Generation for Realistic Sensor Simulation

Ze Yang, Jingkang Wang, Jiageng Mao
{zeyang, wangjk, jmao}@cs.toronto.edu

Motivations

- Generating realistic 3D objects and rendering them at arbitrary viewpoints are important to scale the content creation and sensor simulation for self-driving applications
- Existing 3D generative models focus on the synthetic datasets where the observations are dense, while in the real world the observations are often sparse and noisy
- We focus on the object-level in-the-wild 3D model generation. We experiment some frameworks that could effectively handle the sparse and noisy observations and could reconstruct 3D objects from real images

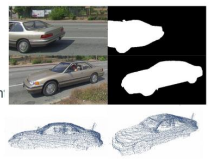
Data Preparation

- Multi-view images for nearby vehicles
- Heuristic occlusion reasoning using LiDAR points
- Segmentation model is employed to generate object masks for each image
- Low-quality objects and images are filtered out

Pandaset dataset

Pandaset is a large-scale autonomous driving dataset:

- 103 driving scenes and 8240 frames
- 6 cameras (front, front-left, left, front-righ right and back) & 2 LiDAR sensors
- 300+ actors (6000 images) after processing

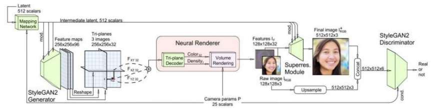


MVMC dataset

Multi-view Marketplace Cars is collected from an online marketplace with images from multiple viewpoints.

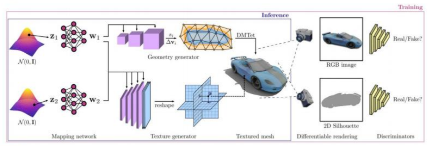
- 576 instances with mask and rough camera pose/intrinsics
- 10+ images per instance

Methods: in-the-wild EG3D and GET3D



EG3D contains 5 key components:

- StyleGAN2-based feature generator
- Tri-plane feature decoder
- Neural volume render
- Super-resolution module
- StyleGAN2 discriminator



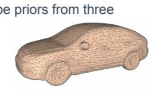
GET3D contains 4 key components:

- Geometry / texture generators
- DMTet representation for explicit triangle mesh extraction
- Differentiable rendering module
- Geometry / texture discriminators

Incorporate shape priors for GET3D

To better handle in-the-wild data, we incorporate shape priors from three perspectives:

- CAD priors (vertex field: Triplane -> vertex offset)



$$\mathbf{V} = \mathbf{V}_{\text{CAD}} + \mathbf{V}_{\text{offset}}$$

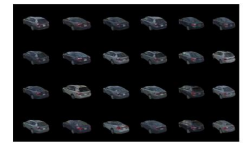
$$E_{\text{sym}}(\mathbf{V}) = \text{Chamfer}(\mathbf{V}, \mathbf{V}_{\text{flip}}) \qquad E_{\text{lap}}(\mathbf{V}) = \sum_i \|\delta_i\|_2^2$$

$$E_{\text{normal}}(\mathbf{V}) = \frac{1}{N_{\mathbf{F}}} \sum_{\mathbf{f} \in \mathbf{F}} \sum_{\mathbf{f}' \in \mathcal{N}(\mathbf{f})} \|\mathbf{n}(\mathbf{f}) \cdot \mathbf{n}(\mathbf{f}')\|_2^2 \qquad E_{\text{edge}}(\mathbf{V}) = \frac{1}{N_{\mathbf{E}}} \sum_{\mathbf{v} \in \mathbf{V}} \sum_{\mathbf{v}' \in \mathcal{N}(\mathbf{v})} \|\mathbf{v} - \mathbf{v}'\|_2^2$$

- Regularization: (a) symmetric chamfer distance, (b) normal consistency loss, (c) average edge length loss, (d) Laplacian loss,

Results - Appearance

GET3D on MVMC

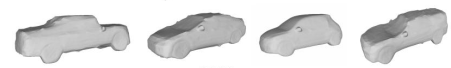


GET3D on Pandaset

EG3D on MVMC

EG3D on Pandaset

Results - Geometry



GET3D

EG3D

Conclusion

- 3D generation in more challenging in real world due to different noise sources and sparse viewpoints
- EG3D & GET3D can generate a variety of real vehicles in the wild although the quality is worse than using synthetic datasets
- EG3D geometries contain more fine-grained details and noise
- GAN training in real world dataset is more unstable and sensitive to the parameter choices