Benchmarking Text to 3D Pipelines Against Novel Pipeline

Robinson Vasquez
University of Central Florida
ro073916@ucf.edu

Michael Miller
University of Central Florida
michael.miller3@ucf.edu

Lam Nguyen
University of Central Florida
la815794@ucf.edu

Michael Ishak
University of Central Florida
mi565439@ucf.edu

Mehrab Mehdi Islam
University of Central Florida
la815794@ucf.edu

February 22, 2025

Abstract

3D models are used in everything from movies, simulation, gaming, art, and industrial manufacturing. To manually create them requires a high investment in time and training. Also for 3D applications such as gaming or movies, each and every unique single object or landscape has to normally be created by a human being and/or purchased as an asset from an asset store. This can be very time consuming and/or expensive as landscapes or rooms can require hundreds or thousands of objects. Different models have been proposed to produce a 3D model from an image and/or text. This paper is intended to benchmark the different methods against our novel method which is to use a diffusion model and combine it with Flash3D to be able to generate a 3D model from a text input. We will then optimize this novel model and compare it to pre-existing methods.

1 Statement of the Problem

The creation of 3D assets is a very time consuming process that requires a large amount of computing resources compared to image production or even video production. Incorporating machine learning models into the workflow can greatly speed up the process either by providing a base 3D model that can be improved and finetuned by a 3d artist or to even allow the 3D artist/modeler to not have to spend any time on the creation of the object at all or to have to purchase from an asset store.

2 Related Work

One of the main technologies that will be used is called Flash3D, which is a model that converts a single image to GSPLAT which is is a differentiable rasterization of 3D Gaussians. Another technology that involves image to 3D is called Splatt3R which is a zero-shot gaussian splatting from image pairs. To create images from text, Deepfloyd will be used which is a low cost diffusion model. Finally, another technology that is involved is known as ControlNet which can be combined with a stable diffusion model to add extra controls on the stable diffusion process such as the ability to specify human poses, copying the composition from another image, and to turn scribble into a professional image.

3 Technical Approach

Our approach involves taking a text prompt as input running it through a diffusion model to produce a single image, using a synthesizer to create multiple views of the image and then putting those images through a Gaussian splat model to reconstruct a 3D model.

1. Text prompt into Diffusion Model
2. Image
3. Image into Canny or Depth Map Extractor
4. Canny/Depth Map
5. Canny/Depth Map into Multi-view Synthesizer
6. Multiple Images
7. Gaussian Splatting Model
8. 3D Model

4 Experiments

The pipeline will be created and 3D models will be generated. This pipeline will then be benchmarked against GaussianDreamer, Large Multi-View Gaussian Model, and Magic3D which are 3D generation algorithms. Then we will test different methods to improve on performance compared to these models.

References

1. LGM: Large Multi-View Gaussian Model for High-Resolution 3D Content Creation, Tang et al, 2024. `https://arxiv.org/pdf/2402.05054`
2. GaussianDreamer: Fast Generation from Text to 3D Gaussians by Bridging 2D and 3D Diffusion Models, Yi et al, 2024. `tps://arxiv.org/pdf/2310.08529`
3. Flash3D: Feed-Forward Generalisable 3D Scene Reconstruction from a Single Image. Szymanowicz et al, 2024. `https://arxiv.org/pdf/2406.04343v1`
4. Speedy-Splat: Fast 3D Gaussian Splatting Sparse Pixels and Sparse Primitives. Hanson et al, 2024. `https://arxiv.org/pdf/2412.00578v1`
5. Splatt3R: Zero-shot Gaussian Splatting from Uncalibrated Image Pairs, Smart et al 2024. `https://arxiv.org/pdf/2408.13912`
6. Adding Conditional Control to Text-to-Image Diffusion Models, Zhang et al 2023. `https://arxiv.org/pdf/2302.05543`
7. DeepFloyd IF, DeepFloyd Lab at StabilityAI. `https://github.com/deep-floyd/IF`