Introduction of Gaussian Function into Rendering

1. Gaussian Splatting

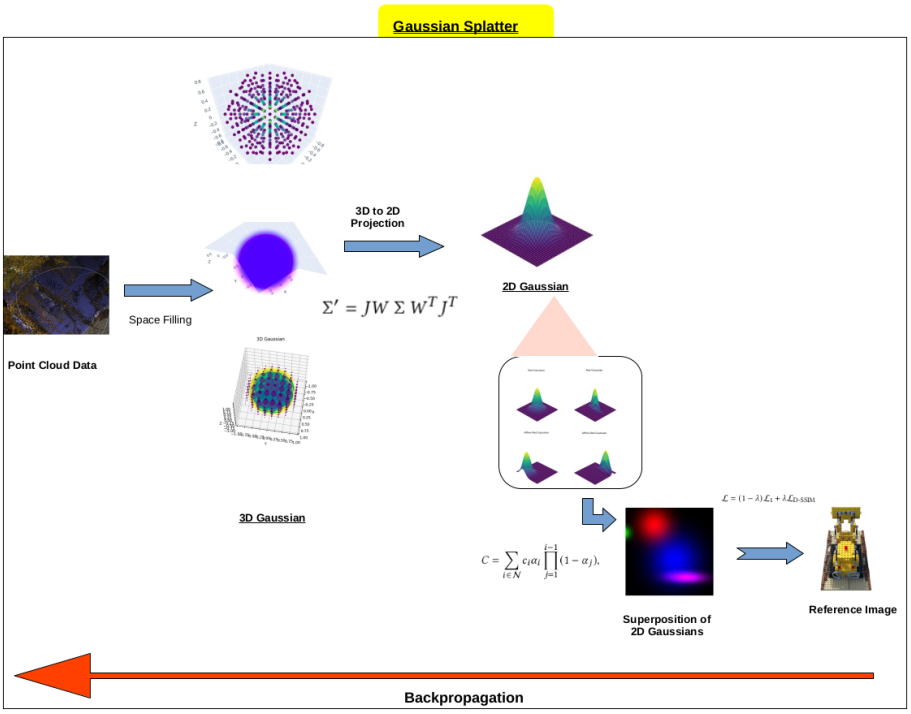

Figure 1: Gaussian Splatting Flow Diagram

1.1. Real time training and rendering

Gaussian functions are initialized where the point cloud points exits. They look like expanding point clouds. Gaussian splatting is fast but has its visual defects [Link]

1.2. Approximating Image with 2D Gaussian

Gaussians move around on image plane to approximate a target image [Link]

1.3. Comparison of Splat Numbering

1.3.1. 100 Splats Small number of Gaussians are slow at approximating image [Link]

1.3.2. 500 Splats Larger number of Gaussians are faster at approximating image [Link]

1.4. Approximating Image with 3D Gaussian

Gaussians initialized in 3D space, and then projected onto 2D image space can approximate an image [Link]

1.5. Comparison of Output

1.5.1. Lego Bulldozer: Train vs Test Gaussian Splatting works great on training (seen) dataset [Link]

Gaussian Splatting works not that great on testing (unseen) dataset [Link]

1.5.2. Train: Train vs Test Gaussian Splatting works great but far away objects look fuzzy on training dataset [Link]

Same is the case for testing dataset [Link]

2. DreamGaussian (Making Gaussian splatting fast with pretrained diffusion models)

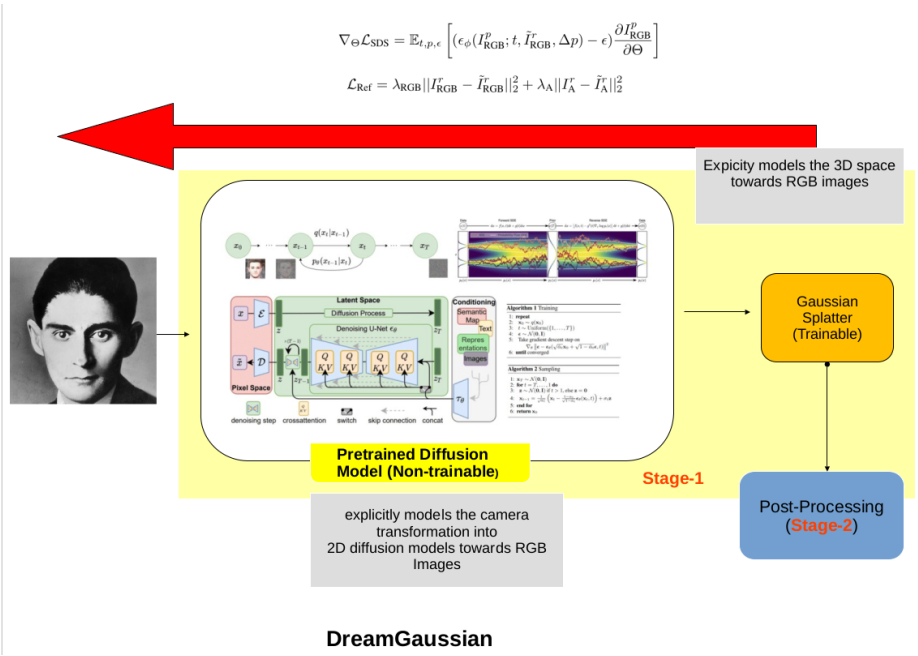

Figure 2: DreamGaussian Flow Diagram

2.1. Comparison of Stage-1 and Stage-2 Output

Stage-1 (while being the main pipeline) leads to fuzzy 3D reconstruction, while Stage-2 improves the finer details, but still results are bad [Link]

Improving iterations of Stage-1 and Stage-2 donot lead to significantly better results [Link]

3. Future Works

- Nerf without parameters

Figure 3: Bulldozer Reconstructed

Figure 4: Desk Reconstructed

- Neuralangelo (its individual componets have been worked upon)

- Improving point cloud representing with Zero: 1-2-3

Code Reference

- Code [Link]