DLCV HW3 Report

B11901040 項達均

December 4, 2024

1

a. Gaussian Splatting represents a 3D scene with Gaussians, which is described by its shape, position, color, and transparency. Before training, point clouds are estimated using the Structure from Motion (SfM) method and then converted into Gaussians. During training, the gaussians are rasterized into images and compared to the ground truth. The Gaussian parameters are then updated with stochastic gradient descent.

b. 3D Gaussian Splatting renders a scene through a set of gaussians, whereas NeRF models a scene by predicting the color and density of each 3D point. Since NeRF requires dense sampling and neural network evaluations for every 3D point along a ray, it is slower and more computationally intensive. On the other hand, it is also more detailed and realistic.

c. The most important part of Gaussian Splatting is its representation of a scene with a set of Gaussians. This significantly reduces the number of trained parameters and thus improves training and inference speed with only a slight trade-off in performance. This allows real-time rendering of scenes, which is important in fields such as VR and AR.

2

The scene is initialized as a set of point clouds using the SfM method, which estimates the 3D structure of a scene from a set of 2D images. Then, each point is converted to a 3D Gaussian parameterized by its position, shape, size, color, and transparency. Given the camera's view angle, we can project the 3D Gaussian onto the 2D image. It is then compared to the ground truth image and the L1 and SSIM loss is calculated. Using this loss, we can update the parameters of the Gaussians to create a more accurate scene with stochastic gradient descent. Automated densification and pruning is also applied to split larger Gaussians and clone smaller Gaussians. This helps the Gaussians model fine-grained details and remove unnecessary Gaussians.

3

Meaning of different metrics:

- PSNR estimates the quality of the generated image. The mean square error between the generated and ground truth images is computed, and the PSNR can be calculated with

$$\mathrm{PSNR}(x, y) = 10 \log \left(\frac{c \, \mathrm{Max}(y)}{\mathrm{MSE}(x, y)} \right) \tag{1}$$

- SSIM estimates the structural similarity between two images. It compares the brightness, contrast, and structure of the images multiplied together.

- LPIPS measures the similarity of two images by computing the distance of the features extracted from neural networks.

Table 1: Evaluation Metrics for Different Settings

Settings			Metrics			
Iterations	**Init Position LR**	**Scaling LR**	**PSNR**	**SSIM**	**LPIPS (vgg)**	**Number of Gaussians**
30000	0.00016	0.005	33.40	0.96	0.13	378023
30000	0.0016	0.05	21.78	0.79	0.46	52023
120000	0.000016	0.001	35.85	0.98	0.08	410476

Here's the training results under different settings: In my experiments, more iterations consistently lead to better results. The number of Gaussians are also highly correlated with PSNR and SSIM score, suggesting that more Gaussians provide a more detailed representation of the scene. We also found that small learning rates performs best in PSNR and SSIM.

4

The position of 3D Gaussians are sampled uniformly in the range of -1.3 and 1.3. Colors are initialized with random spherical harmonics and converted to RGB values. The orientation of the 3D Gaussians are initialized to be zero. The training results using the best setting from part 3 are

Table 2: Evaluation Metrics for Different Settings

Settings			Metrics			
Iterations	**Init Position LR**	**Scaling LR**	**PSNR**	**SSIM**	**LPIPS (vgg)**	**Number of Gaussians**
120000	0.000016	0.001	10.25	0.30	0.64	241603

Using randomly initialized Gaussians in place of the estimated point clouds from the SfM method, the model performs significantly worse because of the lack of meaningful information derived from the 2D images using SfM.