2024 한양대학교

SW/ICT/AI 종합학술대회

[분야 : 산학연구개발/연구인턴]

제목: Gaussian splatting을 이용한 효율적인 실시간 렌더링 알고리즘 개발

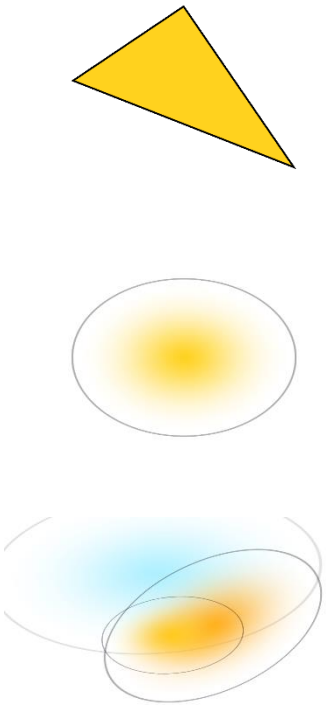
정윤성(21학번)

2024. 11. 21

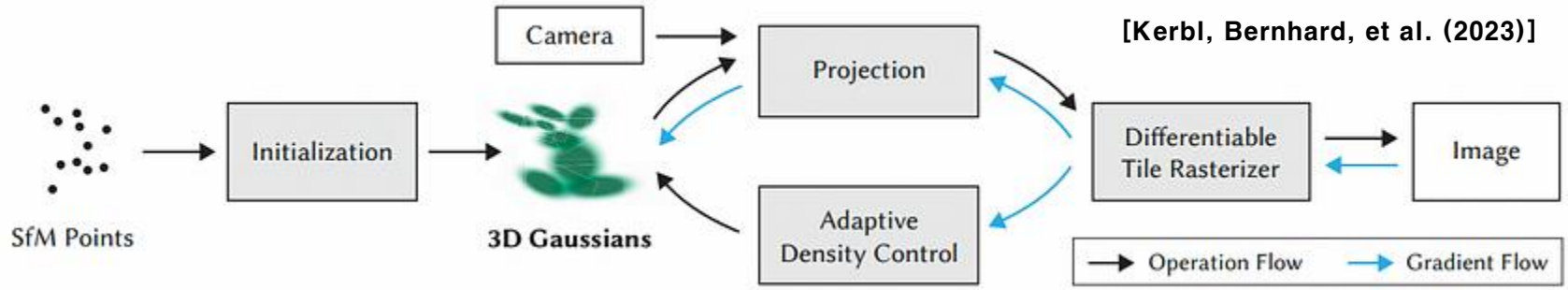
한양대학교 ERICA 소프트웨어융합대학 / 한양대학교 ERICA SW중심대학사업단

Introduction

What is 3D Gaussian Splatting?

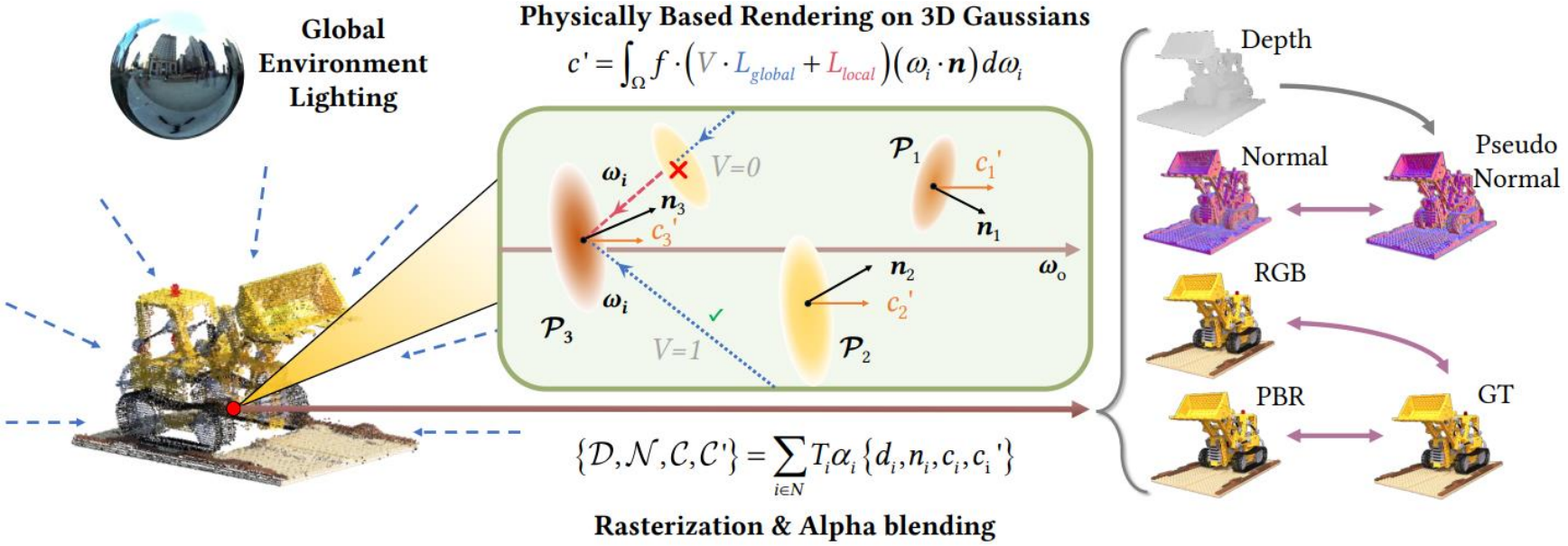




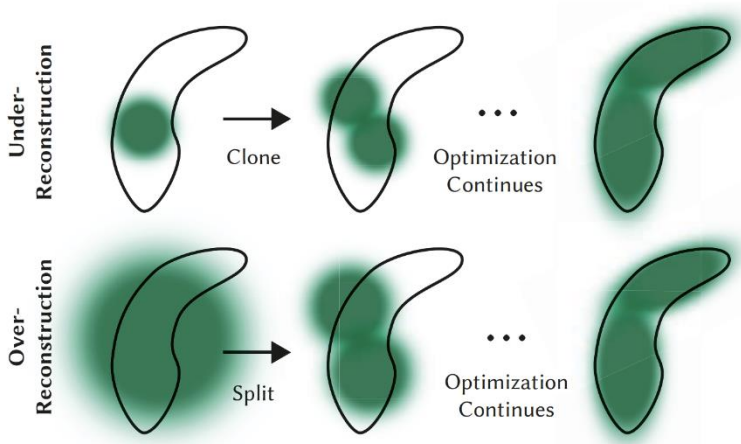


[Kerbl, Bernhard, et al. (2023)]

Introduction



[Gao, Jian, et al.(2023)]



[Kerbl, Bernhard, et al. (2023)]

문제점:

1. 결정론적 방식의 특성
2. 경로 추적에서의 샘플링 및 계산 비용 증가
3. 병렬화의 어려움

=> MCMC를 도입하여 씬 내 가우시안들을 밀집화하는 전략을 적용

1. Large Gaussian 의 경우, 1.6배 작은 2개의 Gaussian 으로 나눔 (split)
2. Small Gaussian 의 경우, gradient 방향으로 복제 (clone)
3. Opacity 가 threshold 이하인 경우 삭제 (prune)

Method

Markov Chain Monte Carlo (MCMC)

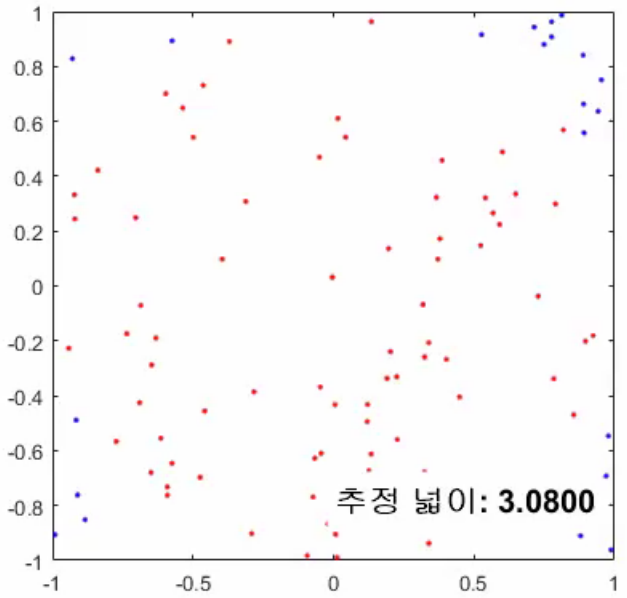
: 샘플링 방법 중 하나

- **Markov Chain**: 어떤 상태에서 다른 상태로 넘어갈 때, 바로 전 단계의 상태에만 영향을 받는 확률 과정
- **Monte Carlo**: 무작위 시뮬레이션을 사용해 문제를 해결하는 기법

```
// compute new opacity
opacity_new[idx] = 1.0f - powf(1.0f - opacity_old[idx], 1.0f / N_idx);
```

opacity_new 값은 이전의 opacity_old 값과 확률적 분포를 기반으로 한 계산(powf(1.0f - opacity_old[idx], 1.0f / N_idx))을 통해 결정된다.
N_idx: 샘플링된 값에 따라 결정되는 비율. opacity의 변화를 조절
-> 이 값이 확률적 요소로 작용

반지름이 1인 원의 넓이를 근사적으로 계산하는 Monte Carlo 시뮬레이션



Stochastic Gradient Langevin Dynamics(SGLD)

[Welling, Max, and Yee W. Teh.(2011)]

: 경사하강법(SGD) + 노이즈. 경사 하강법(SGD)을 확률적으로 변형하여, 확률 분포에 따른 샘플을 얻는 방법

$$\theta_{t+1} = \theta_t - \eta_t \nabla_\theta \log p(\theta) + \sqrt{2\eta_t}\,\epsilon_t$$

- $\theta\, t$: 3D Gaussians의 파라미터
- ηt : 학습률
- $\triangledown\theta \log p(\theta)$: 목표 분포 $p(\theta)$의 로그에 대한 기울기
- $\sqrt{2\eta_t}$: 노이즈의 무작위성 크기 결정. 커질수록 더 넓은 범위를 탐색
- ϵt : 무작위 노이즈

Method

사용한 함수

1. relocate_gs

- 설명: dead_mask를 통해 죽은 점들을 찾고, 살아있는 점들로부터 새로운 값을 샘플링하여 죽은 점들을 갱신한다.
- 작동 방식:
    - dead_mask: 불투명도(opacity)가 너무 작은 점들을 dead_mask로 표시한다.
    - self._alives: 살아있는 점들에서 새로운 점들을 샘플링한다.
    - 새로운 값들(opacity, scaling, normal)을 기존 dead_indices 위치에 업데이트한다.

2. add_gs

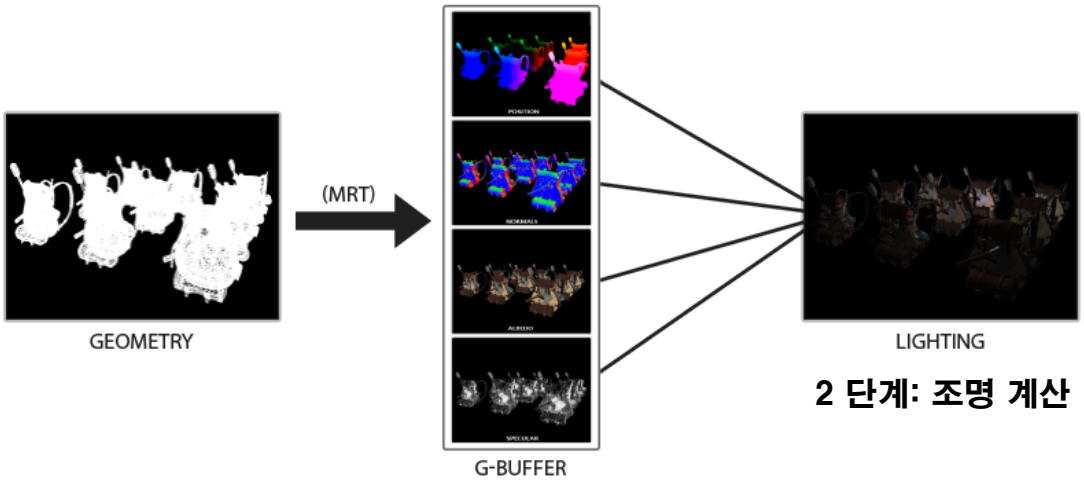
- 설명: 현재 점들의 개수를 기준으로 기존 3DGS 가우시안 수에 맞추어 새로운 3D 점을 추가한다.
- 작동 방식:
    - Current_num_points : 현재 3D 점들의 수를 구한다.
    - 목표하는 가우시안의 개수를 계산하고, 부족한 점들을 샘플링하여 새로 추가한다.
    - densification_postfix : 새로운 점들이 추가된 후 후처리한다.

3. _alives

- 설명: 주어진 probs 에 기반해, num 개의 샘플을 무작위로 선택한다.
  확률 분포에서 샘플링을 통해 선택된 인덱스들은 반환한다.
- 작동 방식:
    - torch.multinomial : 확률 분포에 따라 무작위로 샘플을 뽑는다.
    - sampled_idxs가 주어지면, 해당 인덱스에 대한 샘플링 비율을 계산하여 반환한다.

Method

Deferred Rendering



1 단계: 기하학적 정보 저장

2 단계: 조명 계산

장점:

1. 복잡한 씬에서의 효율적인 조명 처리
   : 조명 계산을 한 번에 처리함으로써, 많은 수의 조명이 있을 때도 성능 저하를 최소화할 수 있다.

2. 다양한 조명 효과를 쉽게 적용할 수 있다.

실제 코드:

```
# Geometry Pass
gbuffer = geometry_pass(viewpoint_camera, pc, pipe, bg_color, scaling_modifier)

# Lighting Pass
features = lighting_pass(viewpoint_camera, pc, dict_params, gbuffer)

# Precomputed values for rasterization
scales = pc.get_scaling if not pipe.compute_cov3D_python else None
rotations = pc.get_rotation if not pipe.compute_cov3D_python else None
cov3D_precomp = pc.get_covariance(scaling_modifier) if pipe.compute_cov3D_python else None
shs = pc.get_shs if override_color is None else None
colors_precomp = override_color if override_color is not None else None
```

⇒대규모 씬의 경우 포인트의 밀도가 높아 광선을 샘플링하고 각 포인트에서 PBR을 수행할 때 최적화 속도가 느리기 때문에 이를 deferred rendering을 통해 완화하였다.

Experiments

Results

NeRF Synthetic Dataset Ficus (0.2fps -> 1fps)
35.01(PSNR)/0.99(SSIM)/0.01(LPIPS) -> 35.26(PSNR)/0.99(SSIM)/0.01(LPIPS)



Conclusion

Limitations and Future work

한계:

1. PSNR의 증가폭이 작다.
2. 프레임 속도가 0.2fps에서 1fps로 약 5배 증가하였지만, 실시간 렌더링의 최소 요건인 30fps에는 못 미친다.

⬇

가장 중요한 프레임 속도의 개선을 위해서는 입사광 예측을 신경망으로 처리하여 렌더링 속도를 높이는 전략이 필요하다. 신경망을 통한 예측은 기존의 수학적 계산을 줄이고, 이를 통해 실시간 렌더링을 가능하게 할 수 있을 것이다.

향후 계획:

1. 입사광 예측: 신경망을 이용한 입사광 모델링
   - 신경망을 이용해 각 표면의 점에 대해 입사광의 양과 방향을 예측한다.
   - 이후 물체의 표면에 도달하는 빛의 양과 그 분포를 모델링하고 이를 통해 각 점에서의 입사광을 실시간으로 계산한다.

2. 리라이트 및 간접광 계산: 몬테 카를로 샘플링을 통한 레이 트레이싱
   - 레이 트레이싱을 할 때 몬테 카를로 샘플링을 사용하여 이 광선들이 물체와 상호작용하는 경로를 무작위로 샘플링한다.
   - 이때 몬테 카를로 샘플링은 여러 개의 광선 샘플을 생성하여, 이들이 어떻게 물체와 상호작용하는지 시뮬레이션한다.
   - 이후 기존에 계산된 입사광 필드와 몬테 카를로 샘플링을 통해 리라이트 한다.

References

Kerbl, Bernhard, et al. "3D Gaussian Splatting for Real-Time Radiance Field Rendering." ACM Trans. Graph. 42.4 (2023): 139-1.

Gao, Jian, et al. "Relightable 3d gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing." arXiv preprint arXiv:2311.16043 (2023).

Kheradmand, Shakiba, et al. "3D Gaussian Splatting as Markov Chain Monte Carlo." arXiv preprint arXiv:2404.09591 (2024).

Welling, Max, and Yee W. Teh. "Bayesian learning via stochastic gradient Langevin dynamics." Proceedings of the 28th international conference on machine learning (ICML-11). 2011.