

Deblurring 3D Gaussian Splatting

- https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/07539.pdf
- ECCV2024

概要

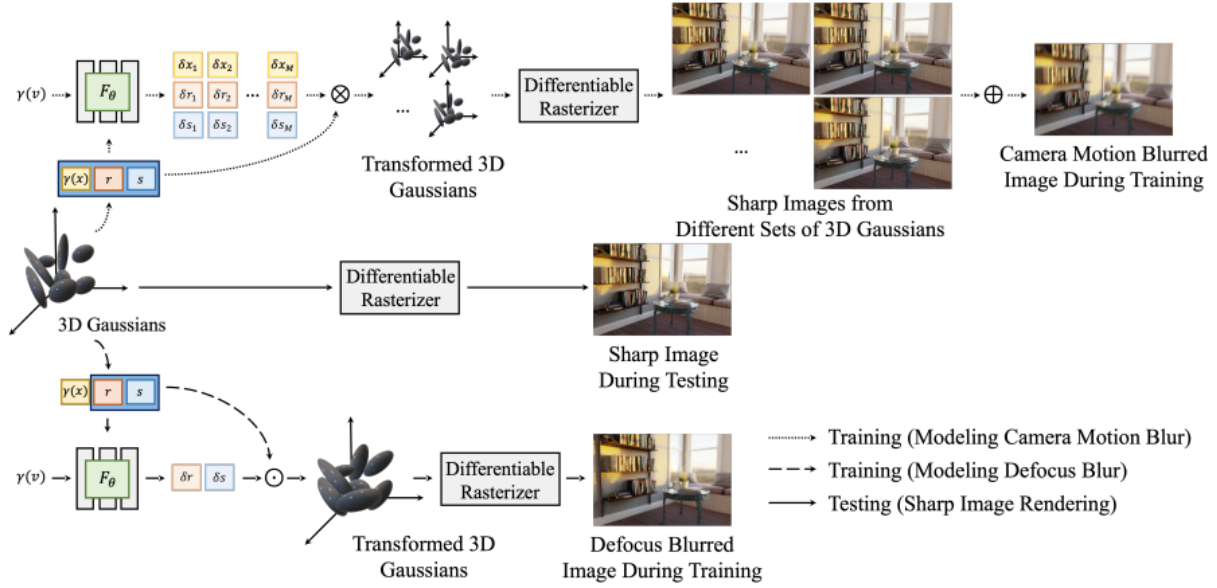
- Deblurring 3D-GSを提案
- 3D-GSのdeblurringアルゴリズムとしてこの論文が初
- 3DGSの後ろに小さいMLPをつける
- このMLPにblurringの過程を学習させる
- こうすることでblur imagesでgsを学習するとMLPがblurringの部分を学習するので推論時にMLPを使わずに3DGSのレンダリングだけやるとdeblur imagesが得られる

related work

- Deblur-NeRF
- DP-NeRF
- PDRF
- どれもNeRF上のdeblurで3DGSはまだない

3DGS

- 全体図は以下



- MLPはdefocus blurを学習するものとcamera motion blurを学習するものを用意

defocus blur modeling

- MLPを使用してrotation, scaleをずらす

$$\begin{aligned}
 (\partial r_j, \partial s_j) &= \mathcal{F}_\theta(\gamma(x_j), r_j, s_j, \gamma(v)) \\
 \hat{r}_j &= r_j \cdot \min(1.0, \lambda_s \partial r_j + (1 - \lambda_s)) \\
 \hat{s}_j &= s_j \cdot \min(1.0, \lambda_s \partial s_j + (1 - \lambda_s))
 \end{aligned} \tag{1}$$

- γ はpositional encoding

$$\gamma(p) = (\sin(2^k \pi p), \cos(2^k \pi p))_{k=0}^{L-1}$$

- 1.0とのminをとっているので r, s はそれぞれ必ず大きくなることが保証されていてこれにより defocus blurを表現できる
- $G(x_j, r_j, s_j) \rightarrow G(x_j, \hat{r}_j, \hat{s}_j)$

camera motion blurring

$$\begin{aligned}
 \{(\partial x_j^{(i)} \partial r_j^{(i)}, \partial s_j^{(i)})\}_{i=1}^M &= \mathcal{F}_\theta(\gamma(x_j), r_j, s_j, \gamma(v)) \\
 \hat{x}_j^{(i)} &= x_j + \lambda_p \partial x_j^{(i)} \\
 \hat{r}_j^{(i)} &= r_j \cdot \partial r_j^{(i)} \\
 \hat{s}_j^{(i)} &= s_j \cdot \partial s_j^{(i)}
 \end{aligned} \tag{2}$$

英語

- pave : 舗装する
- impede : 妨げる
- versatility : 多用途性
- intermingle : 混ざりあう
- harshly : 厳しく