

GaussianSR: 3D Gaussian Super-Resolution with 2D Diffusion Prior

ソース

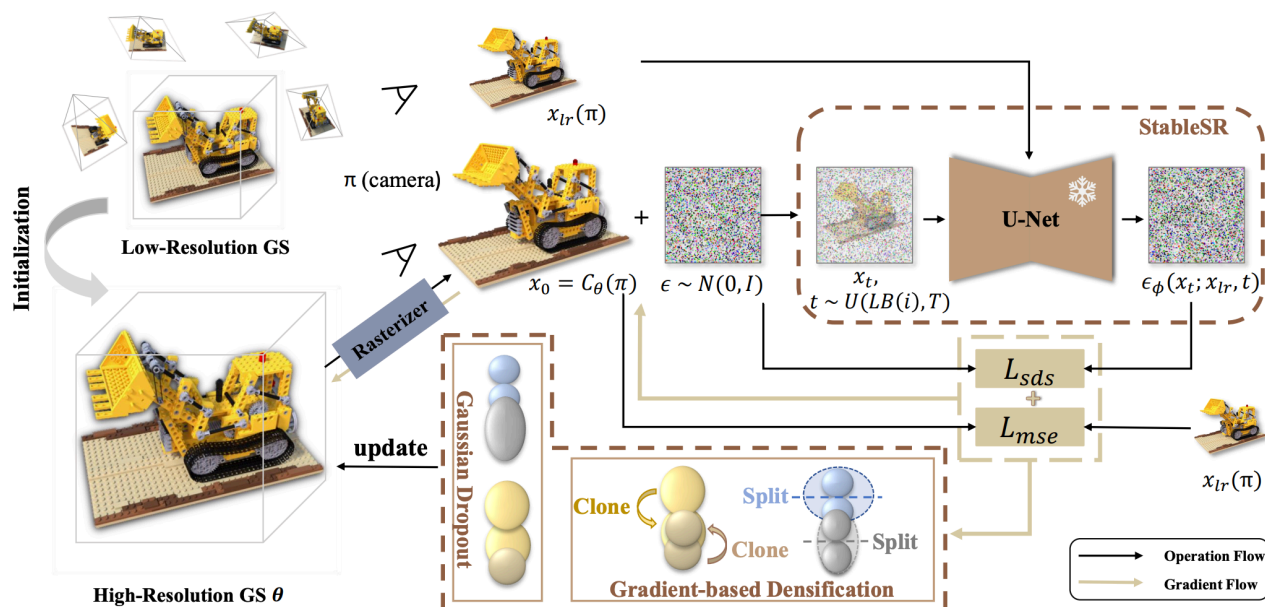
- <https://arxiv.org/pdf/2406.10111>
- <https://chchnii.github.io/GaussianSR/>

概要

- タスクはHRNVS(High Resolution Novel View Synthesis)
- StableSR(diffusionベースの2DSR model)を使ってdistillation
- diffusionでdistillationするとgaussianが余分に増えることが判明したのでdropoutアルゴリズムを導入
- dropoutアルゴリズムは既存のgaussianを消すのではなく splitするときに抑制するアイデア

GaussianSR

- 図を見れば全てわかる



- low resolutionでGS学習

- それを初期状態としてHigh Resolution GSを作成
- Renderingにパラメータ θ を含む
- そのため、HRの画像 x_0 はカメラ情報を π とすると $x_0 = C_\theta(\pi)$ でrenderingされる
- renderingで得られたHR画像 x_0 にノイズを加えて x_t を得る
 - diffusionのforward process
 - t は一様分布からサンプリング
- 得られた $x_t, t, x_{lr}(\pi)$ を入力としてU-Netでdiffusionのreverse processでノイズを出力
 - $x_{lr}(\pi)$ はlow-resolution GSでrendering
- 出力されたノイズと与えたノイズでlossを取る
- x_0 をダウンサンプリングして $x_{lr}(\pi)$ とlossを取る
- このlossから最適化
- gaussianの増加を抑制するためにdrop outアルゴリズムある
- 以上が全てだが、詳しくは以下

3DGS Super-Resolution with SDS Optimization

- SDS : Score Distillation Sampling
- 3DGSでは以下の式を満たすGaussianを2つに分割

$$|g| = \frac{1}{M} \sum_{M_i=1}^M \sqrt{\left(\frac{\partial \mathcal{L}_{M_i}}{\partial \mu_{ndc,x}^{k,M_i}} \right)^2 + \left(\frac{\partial \mathcal{L}_{M_i}}{\partial \mu_{ndc,y}^{k,M_i}} \right)^2} > \tau_{pos}$$

- HR画像のrenderingではパラメータ θ を使って $C_\theta(\pi)$ とする
- 論文中では以下のように説明されている
 - C is the differentiable rendering function for the high resolution image at the given view point π
- 実際にパラメータ θ を使ってどのようにrenderingが計算されているかは一切記載されていない
- コードも公開されていないのでわからない
- このときdistillationのlossについては以下

$$\nabla_{\theta} \mathcal{L}_{SDS}(\phi, C_{\theta}) = \mathbb{E}_{t, \epsilon} \left[w(t) (\epsilon_{\phi}(x_t; x_{lr}, t) - \epsilon) \frac{\partial x}{\partial \theta} \right]$$

- ϕ は事前学習済みのSR diffusion modelのパラメータ
- ϵ_{ϕ} はU-Net
- $w(t)$ はa weight function of different noise levels
- さらに画像のlossも加えて全体のlossは以下になる

$$\begin{aligned}\mathcal{L}_{MSE} &= \|Downsample(x_0) - x_{lr}\| \\ \mathcal{L} &= \mathcal{L}_{MSE} + \lambda \mathcal{L}_{SDS}\end{aligned}\tag{1}$$

Diffusion Timestep Annealing

- diffusionでdistillationするとgaussianを増やしすぎてしまう
- 通常のdiffusionでは各iterationで t を一様分布 $\mathcal{U}(1, T)$ からサンプリングする($t \sim \mathcal{U}(1, T)$)
- この手法ではiteration i で一様分布の下限を1ではなく $LB(i)$ として以下で定義する

$$LB(i) = T - \frac{i}{N}$$

- よって、 $t \sim \mathcal{U}(LB(i), T)$
- 学習の前半の方に多くノイズが付与されるようにしている

Gaussian Dropout

- Gaussianを増やしすぎないように抑制するもの
- 上述の $|g|$ の値が閾値を超えるものが n 個あったとする
- この集合を $G = \{g_0, g_1, \dots, g_n\} \in \mathbb{R}^n$ とする
- 確率 p でセットされたマスク M を用いて G の部分集合 $G' \in \mathbb{R}^k$ を作成
- このとき選ばれた G' だけ2倍にする
- このdensificationを \mathcal{D} で表すと得られるgaussian \hat{G} は以下

$$\hat{G} = \mathcal{D}(G') + (G - G'), \text{ where } G' = G \cdot M(p), M(p) = \begin{cases} 0 & rand(G) < p \\ 1 & else \end{cases}$$

- よって、定義より $\hat{G} \in \mathbb{R}^{n+k}$

英単語

- scarcity : 希少性
- aforementioned : 前述の
- analogous : 類似した