Paper Digest Meeting

Score Distillation via Reparametrized DDIM

Artem Lukoianov Haitz Sáez de Ocáriz Borde Kristjan Greenewald
Vitor Campagnolo Guizilini Timur Bagautdinov
Vincent Sitzmann Justin Solomon

2024.07.03

Jaihoon Kim

KAIST Visual AI Group

Problem Definition

Goal: Generate high-detail 3D objects from input text prompts.

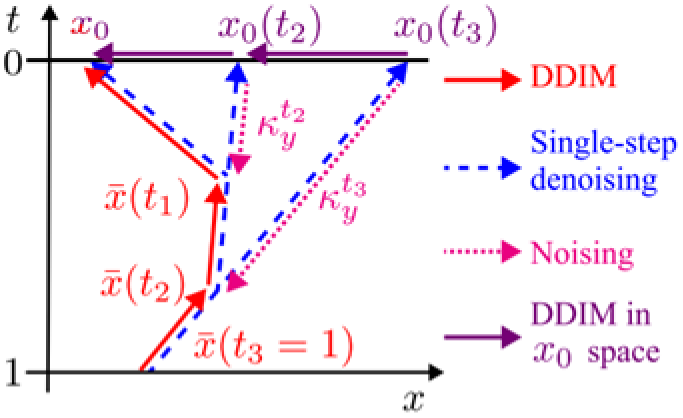
Insight: 2D diffusion models generate realistic, high-detail images whereas SDS produces cartoon-like, over-smoothed images.

a) DDIM (CFG=7.5) b) 2D SDS (CFG=7.5) c) 2D ours (CFG = 7.5)

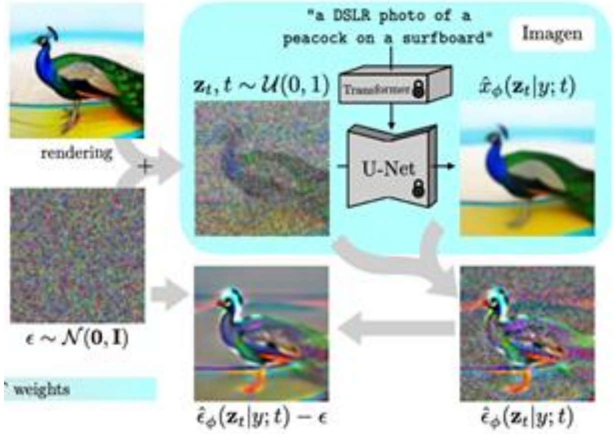
Reparametrization of SDS shows a resemblance to high-variance version of DDIM.
→ A better noise prediction to reduce the variance and improve the output quality.

Key Ideas

DDIM uses the **same predicted noise** for both single-step denoising and noising.
SDS samples random noise at every iteration which unnecessarily increases the variance.



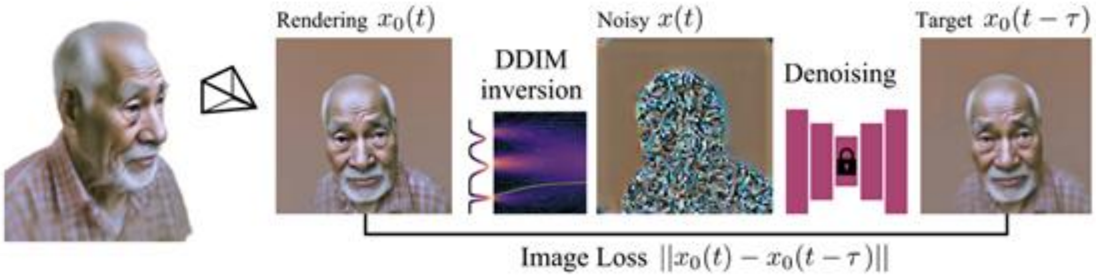
DDIM



SDS

Method

Use DDIM inverted noise to guide SDS as DDIM.



Algorithm 1 Dreamfusion (SDS)

Input: $\psi \in \mathbb{R}^N$ - parametrized 3D shape
$\mathcal{C}$ - set of cameras around the 3D shape
y - text prompt
$g : \mathbb{R}^N \times \mathbb{C} \to \mathbb{R}^{n \times n}$ - differentiable renderer
$\epsilon_\theta^{(t)} : \mathbb{R}^{n \times n} \to \mathbb{R}^{n \times n}$ - trained diffusion model
Output: 3D shape ψ of y

procedure DREAMFUSION(y)
  for i in range(n_iters) **do**
    $t \leftarrow$ Uniform(0, 1)
    $c \leftarrow$ Uniform($\mathcal{C}$)
    $\epsilon \leftarrow$ Normal(0, I)
    $x_t \leftarrow \sqrt{\alpha(t)}g(\psi, c) + \sqrt{1-\alpha(t)}\epsilon$
    $\nabla_\psi \mathcal{L}_{SDS} = \sigma(t)\left[\epsilon_\theta^{(t)}(x_t, y) - \epsilon\right]\frac{\partial g}{\partial \psi}$
    Backpropagate $\nabla_\psi \mathcal{L}_{SDS}$
    SGD update on ψ

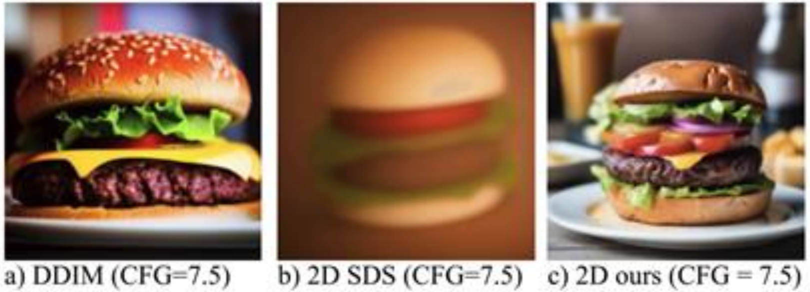
Algorithm 2 Ours (SDI)

Input: $\psi \in \mathbb{R}^N$ - parametrized 3D shape
$\mathcal{C}$ - set of cameras around the 3D shape
y - text prompt
$g : \mathbb{R}^N \times \mathbb{C} \to \mathbb{R}^{n \times n}$ - differentiable renderer
$\epsilon_\theta^{(t)} : \mathbb{R}^{n \times n} \to \mathbb{R}^{n \times n}$ - trained diffusion model
Output: 3D shape ψ of y

procedure OURS(y)
  for i in range(n_iters) **do**
    $t \leftarrow 1 - i/n\_iters$
    $c \leftarrow$ Uniform($\mathcal{C}$)
    $\epsilon \leftarrow \kappa_y^{t+\tau}(g(\psi, c))$
    $x_t \leftarrow \sqrt{\alpha(t)}g(\psi, c) + \sqrt{1-\alpha(t)}\epsilon$
    $\nabla_\psi \mathcal{L}_{SDS} = \sigma(t)\left[\epsilon_\theta^{(t)}(x_t, y) - \epsilon\right]\frac{\partial g}{\partial \psi}$
    Backpropagate $\nabla_\psi \mathcal{L}_{SDS}$
    SGD update on ψ

Experiments

2D Generation - Almost the same as DDIM



3D Generation - Outperform, comparable results to previous methods