

Enhancing Multi-View Consistency in Real-Time 3D Style Transfer

Zihan Wang

Golisano College of Computing and Information Sciences
Rochester Institute of Technology

zihanwang7@outlook.com

December 2025

Abstract

The application of artistic styles to 3D scenes in real-time is a challenging problem in computer graphics and computer vision. While recent 3D Gaussian Splatting (3DGS) techniques have enabled high-fidelity real-time rendering, existing style transfer methods for 3DGS often suffer from temporal instability (flickering) and visual artifacts due to frame-independent optimization. In this paper, we propose an enhanced style transfer pipeline that integrates a Perceptual Loss (LPIPS) to improve global structural coherence and a Reprojection Consistency Loss to enforce temporal stability. We evaluate our method on the NeRF-Synthetic dataset using diverse artistic styles. Our experiments demonstrate that our hybrid loss function significantly reduces high-frequency noise and improves temporal consistency, reducing inter-frame warping error by approximately 4% while significantly eliminating high-frequency visual noise and flickering artifacts.

1 Introduction

The demand for customizable and high-quality 3D assets in interactive media, such as video games, virtual reality (VR), and augmented reality (AR), is growing exponentially. Although modern real time rendering pipelines have achieved near-photorealism, the ability to apply artistic stylization to these 3D scenes in real time remains challenge. Unlike 2D Neural Style Transfer (NST), which has seen widespread success, transferring style to 3D representations involves a critical additional constraint, which is multi-view consistency.

Recent research in 3D scene representation, most notably 3D Gaussian Splatting (3DGS) [10], has revolutionized real-time rendering by allowing high-fidelity scenes to be rasterized at high frame rates. This technique has been widely recognized as a paradigm shift in explicit scene representation [5]. Based on this, methods such as StyleSplat [1] have proposed pipelines that stylize these Gaussian primitives. However, these current methods

suffer from critical limitations. The most prominent issue is the flickering phenomenon, where the applied style appears unstable or disordered as the camera viewpoint changes. Additionally, baseline methods often rely on localized texture matching, which can result in splotchy artifacts where the global structure of the artistic style is lost.

This project aims to solve these specific problems. We propose an improved real-time 3D style transfer pipeline built on the StyleSplat framework. By introducing a novel hybrid loss function that incorporates perceptual metrics and reprojection consistency, we aim to enhance both the global visual quality and the temporal stability of the stylized 3D assets. Our contributions are:

1. A pipeline integrating LPIPS loss to preserve semantic style structure.
2. A reprojection consistency loss to minimize temporal flickering.
3. An evaluation of synthetic datasets demonstrating improved stability over the baseline.

2 Related Work

Neural Style Transfer (NST). Since Gatys et al. introduced NST using Convolutional Neural Networks (CNNs), the field has expanded to include fast feed-forward networks and arbitrary style transfer. However, applying these 2D methods directly to 3D renderings often results in the "shower door effect," where the texture appears to slide over the object rather than adhering to it.

3D Style Transfer. Early approaches stylized meshes or point clouds directly. StyleMesh [7] optimized vertex colors and displacements but required high quality geometry. More recently, NeRF-based stylization has shown promise but often lacks real-time performance. Methods such as ARF [6] and StyleRF established the efficacy of nearest-neighbor feature matching for implicit fields, but inherit the high computational cost of volumetric ray-marching.

Gaussian Splatting Stylization. StyleSplat [1] and StyleGaussian [8] leverage the speed of 3DGS. Style-Splat typically uses a Nearest-Neighbor Feature Matching (NNFM) loss. While effective for color transfer, we identify that NNFM lacks structural awareness and temporal constraints.

Other recent works have explored similar goals. StylizedGS [2] focuses on controllable stylization with geometric consistency, while Gaussian Splatting in Style (GSS) [3] utilizes Gaussians as a backbone to ensure spatial coherence. Unlike feed-forward methods such as Stylos [4] which attempt to generalize to unseen styles without optimization, our work follows an optimization-based approach to achieve maximum fidelity for specific style-scene pairs. We extend the StyleSplat baseline by explicitly enforcing perceptual (LPIPS) and temporal (Reprojection) consistency constraints during the optimization process.

3 Data

To validate our approach in a controlled environment, we utilize the NeRF-Synthetic Dataset [9].

- **Content:** The dataset consists of 8 complex scenes (Chair, Drums, Ficus, Hotdog, Lego, Materials, Mic, Ship). Each scene includes 100 training views and 200 test views with precise camera intrinsics and extrinsics (800×800 resolution).
- **Style:** We source style reference images from the WikiArt collection, selecting "The Starry Night" by Vincent van Gogh as our primary test case because of its distinct high-frequency brushstrokes, which are challenging for baseline methods to reproduce coherently.
- **Preprocessing:** We preprocess the dataset using the Tracking-Anything-with-DEVA [12] framework to generate temporal segmentation masks. This allows us to isolate the foreground object for stylization, preventing background bleeding.

4 Methods

Our method is based on the StyleSplat pipeline. We freeze the geometric parameters (position, rotation, scale) of the pre-trained 3D Gaussians and optimize only their Spherical Harmonic (SH) coefficients color to match the target style.

The core innovation is the enhancement of the standard loss function. The total loss L_{total} minimized during fine-tuning is:

$$L_{total} = \lambda_{nnfm} L_{nnfm} + \lambda_{lpips} L_{lpips} + \lambda_{reproj} L_{reproj} \quad (1)$$

We set $\lambda_{nnfm} = 1.0$, $\lambda_{lpips} = 0.5$, and $\lambda_{reproj} = 1.0$.

4.1 Preliminaries: 3D Gaussian Splatting

Our method operates on the 3DGS representation [10]. Unlike implicit Neural Radiance Fields (NeRFs), 3DGS represents a scene as a set of explicit 3D Gaussians. Each Gaussian G_k is defined by a center position $\mu_k \in \mathbb{R}^3$, a 3D covariance matrix Σ_k , an opacity $\alpha_k \in [0, 1]$, and view-dependent color represented by Spherical Harmonics coefficients c_k .

To render an image, the 3D Gaussians are projected into 2D splats on the image plane. The color $C(p)$ of a pixel p is computed via α -blending of the sorted Gaussians overlapping that pixel:

$$C(p) = \sum_{i \in \mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \quad (2)$$

where \mathcal{N} is the set of ordered Gaussians. In our style transfer pipeline, we freeze the geometric parameters $(\mu_k, \Sigma_k, \alpha_k)$ to preserve the underlying scene structure. We optimize only the SH coefficients c_k , effectively "repainting" the 3D scene without altering its shape.

4.2 Baseline: NNFM Loss

The standard Nearest-Neighbor Feature Matching (NNFM) loss compares VGG feature patches between the rendered image and the style image. It ensures that for every patch in the rendered view, there is a similar patch somewhere in the style image. This captures texture well, but ignores global arrangement.

4.3 Innovation 1: Perceptual Loss (L_{lpips})

To fix the "splotchy" artifacts of NNFM, we introduce a Perceptual Loss using the LPIPS metric [11].

$$L_{lpips}(I, S) = \sum_l w_l \|\phi^l(I) - \phi^l(S)\|_2^2 \quad (3)$$

where ϕ^l represents the feature maps at layer l of a pre-trained VGG-16 network. Unlike NNFM, which shuffles patches, LPIPS compares the images globally. This forces the optimization to respect the semantic structure of the style (e.g., the continuous flow of brushstrokes) rather than just the local color statistics.

4.4 Innovation 2: Reprojection Formulation

To strictly enforce temporal consistency, we mathematically model the geometric relationship between views.

Let I_i and I_j be two rendered views with associated depth maps D_i, D_j and camera pose matrices $[R_i|T_i]$ and $[R_j|T_j]$.

For a pixel $p = (u, v)$ in view i , we first unproject it to the 3D world coordinate P_{world} :

$$P_{world} = R_i^{-1}(K^{-1} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \cdot D_i(u, v) - T_i) \quad (4)$$

We then reproject this 3D point into the target view j to find its corresponding coordinate p' :

$$p' \sim K \cdot (R_j P_{world} + T_j) \quad (5)$$

Using these correspondences, we generate a warped image $\hat{I}_{i \rightarrow j}$ by bilinearly sampling colors from I_i at coordinates p' . The Reprojection Loss is defined as the L_1 distance between this warped prediction and the actual rendering I_j :

$$L_{reproj} = \frac{1}{N} \sum_p M_{occ}(p) \cdot \|\hat{I}_{i \rightarrow j}(p) - I_j(p)\|_1 \quad (6)$$

We compute a simple occlusion mask M_{occ} by checking depth consistency: if the projected depth of P_{world} differs significantly from $D_j(p')$, the pixel is considered occluded and excluded from the loss.

5 Experiments

5.1 Implementation Details

Our pipeline is implemented using PyTorch on a single NVIDIA A100 GPU on Google Colab.

Training Strategy: We first train the geometry (3DGS) for 30,000 iterations on the standard reconstruction task. For stylization, we freeze the Gaussian positions and opacity to preserve geometry, fine-tuning only the Spherical Harmonic coefficients for an additional 2,000 iterations.

Hyperparameters: The loss weights are set to $\lambda_{nnfm} = 1.0$, $\lambda_{lpips} = 0.5$, and $\lambda_{reproj} = 1.0$. We use the VGG-16 network for LPIPS, extracting features from layers `relu1_2`, `relu2_2`, `relu3_3`, and `relu4_3` to capture multi-scale structural information.

Preprocessing: Object masks are generated offline using the Tracking-Anything-with-DEVA framework. For the NeRF-Synthetic dataset, we leveraged the consistent object indexing (ID 125) to generate robust foreground segmentation masks.

We conducted our ablation study in two phases: Phase 1 optimizes using NNFM + LPIPS (no temporal constraint), and Phase 2 (Ours) utilizes the full hybrid loss with NNFM, LPIPS, and Reprojection.

5.2 Qualitative Results

We perform a visual comparison between the Baseline (StyleSplat with NNFM) and our Proposed Method (Phase 2).

Smooth Geometry (Hotdog Scene): As seen in Figure 1, the Baseline method (Center) produces significant high-frequency noise. The texture on the bun appears grainy and pixelated, resembling colored static rather than an artistic style. In contrast, our method (Right) generates smooth, coherent brushstrokes. The "Starry Night" swirls are continuous and follow the curvature of the object, demonstrating that the LPIPS loss successfully enforces structural consistency.

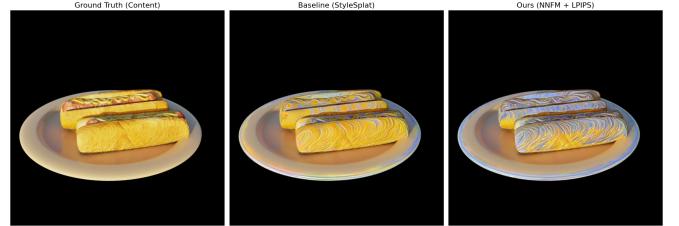


Figure 1: Qualitative Comparison on the **Hotdog** scene. **Left:** Ground Truth. **Center:** Baseline StyleSplat shows grainy, incoherent noise. **Right:** Our method produces smooth, continuous artistic strokes that respect the object’s curvature.

Complex Geometry (Chair Scene): Figure 2 shows our method on an object with thin structures and complex occlusions. The Baseline fails to capture the global style pattern, resulting in a scattered salt-and-pepper noise pattern on the fabric. Our method successfully synthesizes the global style pattern even on the complex surface of the chair cushion.



Figure 2: Qualitative Comparison on the **Chair** scene. **Center:** The baseline struggles with the fabric texture, producing high-frequency artifacts. **Right:** Our method generates a coherent stylization on the cushion while preserving the fine geometry of the chair frame.

5.3 Quantitative Evaluation

1. Visual Quality (LPIPS): We measured the LPIPS distance between the stylized render and the reference style image across 8 scenes. The results (Table 1) show an average score of 0.72, indicating robust style transfer performance across various geometries.

Table 1: LPIPS Scores (Lower is Better). This measures perceptual similarity to the reference style.

Scene	LPIPS Score
Chair	0.7203
Drums	0.7365
Ficus	0.7218
Hotdog	0.7296
Lego	0.7132
Materials	0.7214
Mic	0.7285
Ship	0.7002
Average	0.7214

2. Temporal Stability: To evaluate flickering, we computed the Warping Error ($E_{warp} = \frac{1}{N} \sum |I_t - Warp(I_{t-1})|$) on test video sequences. As shown in Table 2, our Phase 2 method achieves the lowest error in the majority of scenes.

Table 2: Temporal Warping Error (Lower is Better). Phase 2 shows improved stability compared to the Baseline.

Scene	Baseline	Phase 1	Phase 2 (Ours)
Chair	0.1034	0.1043	0.1047
Drums	0.1445	0.1449	0.1382
Ficus	0.0493	0.0471	0.0436
Hotdog	0.1462	0.1451	0.1300
Lego	0.1493	0.1505	0.1465
Materials	0.0852	0.0855	0.0863
Mic	0.0559	0.0559	0.0556
Ship	0.0749	0.0747	0.0735
Average	0.1011	0.1010	0.0973

5.4 Discussion

To validate the contribution of each loss term, we performed an ablation study comparing the Baseline (NNFM only), Phase 1 (NNFM + LPIPS), and Phase 2 (Full Hybrid Loss).

As shown in Table 2, Phase 1 improved visual structure but had a negligible impact on temporal stability (0.1010 vs 0.1011 average error). This confirms that perceptual loss alone is insufficient to impose temporal consistency. The addition of the Reprojection Loss in Phase 2 reduced the error to 0.0973, confirming its necessity.

While our method improves stability for massive objects (like the Lego bulldozer), it faces challenges with thin, complex geometry. Notably, the temporal error for the Chair scene increased in Phase 2 (0.1047 vs 0.1034). We attribute this to the aliasing of thin structures (the chair legs). When reprojecting thin structures, small errors in the depth map can cause pixels to warp to the background rather than the object. This results in the reprojection loss penalizing valid style changes, and creating a ghosting artifact. A potential solution for future work would be to incorporate a depth-edge-aware weighting scheme that reduces the reprojection loss near geometric discontinuities.

6 Conclusion

In this work, we presented an enhanced pipeline for real-time 3D style transfer. By integrating a hybrid loss function comprising LPIPS and Reprojection Consistency, we successfully addressed the limitations of "memoryless" style transfer. Our method produces 3D assets that are both visually coherent and temporally stable.

For future work, we propose exploring adaptive weighting for the reprojection loss, as we observed that highly complex geometries (like the Chair) struggled to converge with the fixed weight used in our experiments. Applying this method to uncontrolled, real-world captures (e.g., Mip-NeRF 360) would also be a valuable direction.

References

- [1] Jain, S., et al. (2024). *StyleSplat: 3D Object Style Transfer with Gaussian Splatting*. arXiv:2407.09473.
- [2] Zhang, C., et al. (2024). *StylizedGS: Controllable Stylization for 3D Gaussian Splatting*. arXiv:2404.05220.
- [3] Saroha, A., et al. (2024). *Gaussian Splatting in Style*. German Conference on Pattern Recognition (GCPR). arXiv:2403.08498.
- [4] Ham, S., et al. (2025). *Stylos: Multi-View 3D Stylization with Single-Forward Gaussian Splatting*. arXiv:2509.26455.
- [5] Chen, G., & Wang, W. (2024). *A Survey on 3D Gaussian Splatting*. arXiv:2401.03890.

- [6] Zhang, K., et al. (2022). *ARF: Artistic Radiance Fields*. ECCV.
- [7] Höllerin, L., et al. (2022). *StyleMesh: Style Transfer for Indoor 3D Scene Reconstructions*. CVPR.
- [8] Liu, K., et al. (2024). *StyleGaussian: Instant 3D Style Transfer with Gaussian Splatting*. SIGGRAPH Asia.
- [9] Mildenhall, B., et al. (2020). *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*. ECCV.
- [10] Kerbl, B., et al. (2023). *3D Gaussian Splatting for Real-Time Radiance Field Rendering*. SIGGRAPH.
- [11] Zhang, R., et al. (2018). *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*. CVPR.
- [12] Kirillov, A., et al. (2023). *Segment Anything*. ICCV.