

WaSt-3D: Wasserstein-2 Distance for Scene-to-Scene Stylization on 3D Gaussians

Dmytro Kotovenko^{1*}, Olga Grebenkova¹, Nikolaos Sarafianos², Avinash Paliwal³, Pingchuan Ma¹, Omid Poursaeed², Sreyas Mohan², Yuchen Fan², Yilei Li², Rakesh Ranjan², and Björn Ommer¹

¹ CompVis @ LMU Munich and MCML

² Meta Reality Labs

³ Texas A&M University

Abstract. While style transfer techniques have been well-developed for 2D image stylization, the extension of these methods to 3D scenes remains relatively unexplored. Existing approaches demonstrate proficiency in transferring colors and textures but often struggle with replicating the geometry of the scenes. In our work, we leverage an explicit Gaussian Scale (GS) representation and directly match the distributions of Gaussians between style and content scenes using the Earth Mover’s Distance (EMD). By employing the entropy-regularized Wasserstein-2 distance, we ensure that the transformation maintains spatial smoothness. Additionally, we decompose the scene stylization problem into smaller chunks to enhance efficiency. This paradigm shift reframes stylization from a pure generative process driven by latent space losses to an explicit matching of distributions between two Gaussian representations. Our method achieves high-resolution 3D stylization by faithfully transferring details from 3D style scenes onto the content scene. Furthermore, WaSt-3D consistently delivers results across diverse content and style scenes without necessitating any training, as it relies solely on optimization-based techniques. See our project page for additional results and source code: <https://compvis.github.io/wast3d/>.

Keywords: 3D Stylization · 3D Gaussian Splatting · NeRF · Style Transfer · Optimization

1 Introduction

Style transfer is a well researched method that allows to create artistic images, which combines content from one input source and style from another. This process has been widely used in 2D images to create visually appealing and unique results. However, as virtual reality and 3D approaches become more prevalent, there is a growing interest in adapting style transfer techniques to work in the three-dimensional space. Incorporating style transfer into 3D visuals opens

* Work done during an internship at Meta.

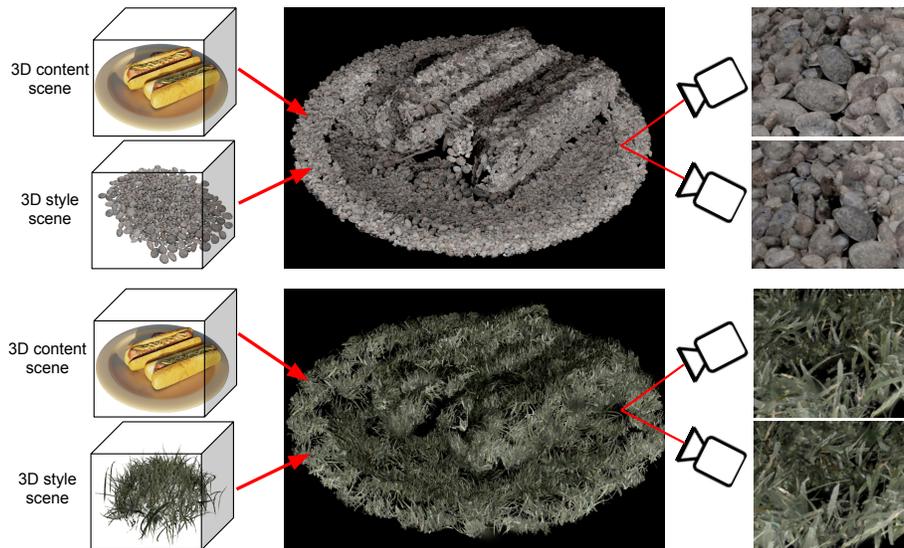


Fig. 1: Our stylization of the hot dog scene using 2 different style scenes: *pebbles* and *grass*. Our method accepts as an input two pretrained Gaussian splattings scenes and merges them together by clustering content scene, selecting best style cluster and finally minimizing Sinkhorn divergence between the pairs. High resolution 3D details are depicted in the last column by rendering the same shape volume from two different viewpoints.

up new possibilities for creating unique and engaging content. By merging the artistic styles of different sources with the three-dimensional aspects of virtual reality, creators can push the boundaries of what is possible in terms of visual storytelling and design.

In computer graphics, stylized models serve as artistic interpretations of 3D objects, often featuring exaggerated or simplified elements reminiscent of cartoon characters, abstract shapes, or low-poly designs. This creative approach extends to 3D game design, where various art styles shape the visual identity of virtual worlds and gaming experiences. Realism and Fantasy Realism aim for lifelike authenticity, while Low Poly art embraces a minimalist aesthetic with simplified geometric shapes, and Cartoon art infuses games with whimsical charm and vibrant colors. These diverse styles utilize different textures and modeling techniques to evoke specific emotions and atmospheres. While currently prevalent in the movie and gaming industries, there is a growing need to democratize these applications, making them easily accessible and usable for consumers.

Prior works in 3D have focused primarily on altering textures, such as applying the color palette or patterns of one object onto another [4, 7, 8, 15, 20, 46]. Another application domain is video, where the main challenge is maintaining temporal consistency across frames [11, 49]. While this can create visually inter-

esting results, there is a need to explore how style transfer can be applied to both textures and geometry in 3D visuals.

Throughout art history, the study of style has encompassed various elements such as color, composition, and, notably, geometry and shape, which are considered crucial aspects in defining artistic identity [2, 9]. In art, the technique of assemblage involves creating three-dimensional compositions on a defined substrate, akin to collage in two dimensions. Often utilizing found objects, assemblage offers a multidimensional approach to artistic expression. An illustrative example of this idea applied to oil paintings can be found in the works of Giuseppe Arcimboldo, who crafted human portraits from seasonal fruits, grains, and vegetables. Another example of this technique is the “Mandolin” sculpture by Pablo Picasso assembling the object from the wood pieces, see Fig. 2. Drawing inspiration from this approach, we propose a method where content object is assembled from various elements of the style scene.

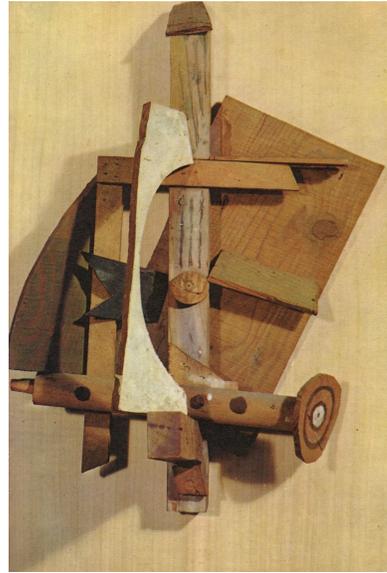
In style transfer, the relationship between content and style representation is one of the central problems. The seminal work by Gatys *et al.* [8] emphasized the optimization process that balances style and content loss to generate images, laying the foundation for subsequent investigations in the field. This paper laid the groundwork for subsequent explorations in the field, which have extended beyond two-dimensional texture transfer into three-dimensional shapes, facilitating the transfer of spatial structures between scenes.

The conventional optimization-based 2D style transfer matches style and content features distribution of the stylized image and style/content image respectively. Following this approach, we initially represent the content and style scenes as two-colored collections of particles using regularized Gaussian splatting (refer to Sec. 4 for details). To smoothly align these distributions, various methods can be used, but we choose to employ Sinkhorn divergence between the distributions of points corresponding to style and content scenes. Sinkhorn Divergence determines how the style scene should be adjusted to match the content scene. However, handling distributions with large sets of points becomes challenging. Therefore, we first segment the content scene into compact shapes and identify the best-fitting style segment. Then, we estimate the Sinkhorn divergence between pairs of style and content clusters, optimizing the selected style cluster to minimize the divergence between them. We opt for the Sinkhorn Divergence since it acts as a “debiased” Wasserstein-2 distance and is easier to tune.

In this study, we introduce a novel method for 3D scene style transfer, with the goal of accurately reproducing the geometry of the style scene within the content scene. Utilizing Sinkhorn divergence to adapt and integrate elements from the style scene, we construct a stylized content scene that preserves high visual fidelity and faithfully captures the geometric details of the style. Our qualitative experiments showcase that our approach outperforms existing methods, particularly in terms of geometric stylization quality.



(a) “Autumn” by Guiseppe Arcimboldo.



(b) “Mandolin” by Pablo Picasso.

Fig. 2: Two illustrations of the assemblage principle in art. The coarse level content can be constructed of local pieces of a style object of a different nature: vegetables and various pieces of wood. We follow a similar idea in our approach.

2 Related Work

Style Transfer Style transfer aims at generating images with the aesthetic style of the given style images while preserving their semantic content. Traditional methods use handcrafted features to simulate styles [14]. Gatys *et al.* [8] introduced neural style transfer in which the content and style representations are obtained as intermediate activations and the corresponding Gram matrices of a pre-trained VGG network [43]. The resulting image is generated by iteratively updating a random input until its content and style representations match the reference ones. Follow-up works [10, 20, 21, 23, 25, 28, 30, 38] have explored alternative style loss formulations to improve the quality of semantic consistency and high-frequency style details. Rather than performing iterative optimization, feed-forward approaches [1, 7, 13, 22, 27, 34, 39, 42, 52] train neural networks that can capture the style information of the style image and transfer it to the input image with a single forward pass. These methods are faster than optimization-based techniques but often yield lower visual quality. Unlike neural style transfer methods that encode statistics of style features with a single Gram matrix, another line of work searches for nearest neighbors and minimizes distances between features extracted from corresponding content and style patches [6, 19, 25, 28]. These methods achieve impressive 2D stylization quality when provided with source and target images that share similar semantics [54].

2.1 3D style transfer

In recent years many papers tried to transfer 2D results in 3D dimensions. The works can be generally split into two groups: with target style geometry and without one.

With the access to the style target geometry Nerf-Tex [3] fills the content scene with patches of the style example. A similar approach was adopted by [45], where the style geometry is predefined using tessellation maps to articulate fine details in 3D geometry. An alternative strategy involves leveraging depth maps alongside the original style image [16]. However, these approaches encounter challenges when generalizing across diverse and intricate scenes with full 360-degree camera rotation, as the target depth is specified only for a single fixed viewpoint.

Concurrently, a separate line of research focuses on modifying only the geometry of shapes without incorporating style textures [24, 41, 48].

For methods lacking access to style geometry, the transformation is attempted based on 2D information using common 3D representations: point clouds [12], meshes [26, 40], NeRFs [5, 29, 33, 47, 51, 54, 55] or Gaussian Splats(GS) [53]. Another approach to provide style information is by incorporating text prompt with desirable changes. Method [5] works using a text prompt indicating name of the artist in whose style the final image will be presented. However, methods that take as an input image or text prompt barely change the geometry of the scenes. In majority of cases such approaches only mimic the colors of the style scene without altering the geometry.

3 Methodology

In this section, we describe the scene representation used throughout the paper, namely regularized Gaussian splattings. Following this, we introduce the Wasserstein-2 distance and its debiased counterpart, the Sinkhorn divergence, to quantify the divergence between two sets of Gaussian splattings. Finally, we outline WaSt-3D stylization algorithm, which is divided into three parts: a) cluster the content scene into disjoint subsets, b) select the best-fitting style cluster for each content cluster using the selection function D , and c) optimize style clusters to align with corresponding content clusters by minimizing the Sinkhorn divergence.

3.1 Background: Gaussian splatting representation

Neural Radiance Fields (NeRFs) [32] have widely been utilized over the past few years as a neural network based representation that implicitly learns the scene density and view-dependent color. This representation allows for high-quality novel view synthesis once the NeRF model is optimized from a set of 2D input images. More recently, 3D Gaussian Splatting (3DGS) [17] has emerged as a powerful alternative that provides significant advantages in terms of optimization

speed and real-time rendering while matching the state-of-the-art quality from NeRF-based approaches. 3DGS achieves this by relying on explicit representation based on 3D Gaussians in contrast to the implicit representation of NeRF. To enable high-quality scene reconstruction, each Gaussian is characterized by a set of learnable parameters such as position $\mathbf{x} \in \mathbb{R}^3$, color $\mathbf{c} \in \mathbb{R}^3$, covariance matrix $\Sigma \in \mathbb{S}\mathbb{O}(3)$ (shape and orientation), scaling $\mathbf{S} \in \mathbf{diag}(\mathbb{R}_+^3)$ and opacity α . The rendering equation for a typical point-based approach is defined as:

$$C = \sum_{i \in \mathcal{N}} \mathbf{c}_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j). \quad (1)$$

Here, \mathcal{N} is the number of ordered points along the ray that are α -blended per pixel and α_i is the blending opacity obtained by evaluating the 2D Gaussian projection at the pixel and multiplying it with the global opacity α .

3.2 Anisotropic Gaussian Splattings Representation

Default 3D Gaussian splattings possess several parameters beyond coordinate and color, including scaling, rotation, and opacity. However, the objective of the training regime is often pixelwise similarity between the rendered image and the target image. This can inadvertently lead to individual Gaussians being stretched, resulting in undesirable needle-shaped artifacts protruding from the surface. This may result in unpleasant visual artifacts when we start splitting scenes into subsets with function D_i . To mitigate this effect, we introduce an anisotropic regularization for the individual Gaussian splattings, aiming to force them into spherical shapes. This is accomplished by minimizing the difference between the largest and smallest scaling components $g_{\mathbf{s}}$ of each Gaussian $g \in \mathcal{G}$:

$$\mathcal{L}_{aniso} = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \left(\max \left(\frac{\max(g_{\mathbf{s}})}{\min(g_{\mathbf{s}})}, r \right) - r \right), \quad (2)$$

where the scalar r determines how much can the largest and the smallest scaling parameter differ. Additionally we want to have Gaussians to have similar scale across the whole image.

$$\mathcal{L}_{unif} = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \|g_{\mathbf{s}} - \hat{g}_{\mathbf{s}}\|^2, \quad (3)$$

where $\hat{g}_{\mathbf{s}} = (s, s, s) \in \mathbb{R}_+^3$ denotes target scale of Gaussians with $s \in \mathbb{R}_+$ being a positive real value indicating equal scale in every dimension. Similar idea has been introduced as a regularization technique in [50]. Moreover, effect of the anisotropic regularization is discussed in the original paper on 3D Gaussian splattings [17].

3.3 3D Style Transfer with Gaussian Splatting

Wasserstein-2 distance. The goal of the style transfer is to blend the style and content of two inputs across various domains such as images, audio, video, text, and 3D scenes. In this paper, we focus on blending two 3D scenes represented with 3D Gaussian splattings. We represent the style and content scenes by two collections of 3D Gaussian splattings, denoted as $\mathcal{G}_{\text{style}}$ and $\mathcal{G}_{\text{content}}$ respectively. The primary goal of stylization is to obtain a shape that locally resembles the style scene while globally resembling the content scene.

From a probabilistic standpoint, we need to ensure that the distributions of the style and content scenes are similar on a local scale. The distributions p_c and p_s of the Gaussians $\mathcal{G}_{\text{content}}$ and $\mathcal{G}_{\text{style}}$ lie on a compact connected Riemannian manifold M with geodesic distance $d : M \times M \rightarrow \mathbb{R}_+$. We can use the Wasserstein-2 distance to find the optimal transport between them $\pi \in \Pi(p_s, p_c)$

$$\Pi(p_s, p_c) := \left\{ \pi \in \text{Prob}(M \times M) : \pi(\cdot, M) = p_c, \pi(M, \cdot) = p_s \right\}. \quad (4)$$

$$\mathcal{W}_2(p_s, p_c) := \left[\inf_{\pi \in \Pi(p_s, p_c)} \iint_{M \times M} d(x, y)^2 d\pi(x, y) \right]^{\frac{1}{2}}. \quad (5)$$

This distance defines how dissimilar style and content distributions are. In practice, solving this problem is computationally intractable, necessitating regularization to make it applicable to the domain of Gaussian splatting in 3D spaces. One approach to encourage spread-out transportation plans π is by adding the entropy term $H(\pi)$ to the Wasserstein-2 distance in Eq. (5), resulting in an entropy-regularized objective function:

$$\mathcal{W}_{2,\gamma}^2(p_s, p_c) := \left[\inf_{\pi \in \Pi(p_s, p_c)} \iint_{M \times M} d(x, y)^2 d\pi(x, y) - \gamma H(\pi) \right]. \quad (6)$$

The entropy-regularized version of our objective function renders it strictly convex and is commonly referred to as the ‘‘Schroedinger problem’’. In practice, this regularization smoothens the problem and prevents overfitting. Varying the parameter γ allows us to control the smoothness of the transport plan between the two distributions. A higher γ value results in a smoother transport plan, where every Gaussian from the style scene is transported to every Gaussian from the content domain, approximating an ‘‘average’’ Gaussian. Conversely, a very small γ value leads to each Gaussian being transported to just one Gaussian from the other domain, making the metric more specific.

We could have utilized Eq. (6) to compute the distance between two distributions and perform gradient flow to map one distribution to another. However, this is not a metric because $\mathcal{W}_{2,\gamma}^2(p_s, p_s) \neq 0$ and $\mathcal{W}_{2,\gamma}^2(p_c, p_c) \neq 0$. To address this, we employ the debiased version of the entropy regularized Wasserstein-2 distance dubbed Sinkhorn divergence:

$$\mathcal{SD}_{2,\gamma}^2(p_s, p_c) := \mathcal{W}_{2,\gamma}^2(p_s, p_c) - \mathcal{W}_{2,\gamma}^2(p_s, p_s) - \mathcal{W}_{2,\gamma}^2(p_c, p_c). \quad (7)$$

For values $\gamma \rightarrow \infty$ objective turns into $\frac{1}{2}MMD^2(p_s, p_c)$, $\gamma \rightarrow 0$ objective turns into $\mathcal{W}_2^2(p_s, p_c)$. See [35, 37], where MMD stands for the Maximum Mean Discrepancy. This divergence between the distribution can be used to compute gradients and update one of the distributions. To estimate this value we use the Sinkhorn iterations method.

Scene partitioning. Estimating optimal transport between distributions containing hundreds of thousands or millions of points is not feasible, even with approximation algorithms. Thus, we tackle this problem by dividing it into smaller chunks and stylizing each part of the content scene separately. Although this may seem limiting, this approach guarantees faithful representation of the content scene by substituting it with small pieces of the style scene. Essentially, we want to ensure that our stylization locally resembles the style scene.

To achieve this, we divide the content scene $\mathcal{G}_{\text{content}}$ into a collection of clusters $C = \bigcup_{i \in \{1, \dots, N\}} C_i$, where each cluster consists of a collection of Gaussians $C_i = \{g | g \in \mathcal{G}_{\text{content}}\}$. Our objective is to transform these clusters while preserving the style and accurately representing the content scene. To achieve this, we first determine which style cluster fits best with each content cluster C_i .

This correspondence function D_i between content clusters and parts of the style scene $\mathcal{G}_{\text{style}}$ is crucial for selecting the distributions p_s and p_c to be matched using the optimal transport plan.

To identify content clusters $C = \bigcup_{i \in \{1, \dots, N\}} C_i$, we perform K-Means clustering on the colored Gaussians. We then determine which style cluster should be replaced with the best-fitting style cluster by formulating a constrained optimization task. We fit each content cluster C_i to the style scene, using only translation $\mathbf{t}_i \in \mathbb{R}^3$, rotation $\mathbf{R}_i \in \mathbb{SO}(3)$ ($\mathbb{SO}(3)$ is special orthogonal group of order 3), and scaling $\mathbf{S}_i \in \mathbf{diag}(\mathbb{R}_+^3)$:

$$\begin{aligned} \min \sum_{g \in C_i} \{ \min \| \mathbf{S}_i \mathbf{R}_i (g_{\mathbf{x}} - \mathbf{t}_i) - g'_{\mathbf{x}} \|_2 \mid g' \in \mathcal{G}_{\text{style}} \} \\ \text{s.t. } \mathbf{R}_i \in \mathbb{SO}(3), \mathbf{S}_i \in \mathbf{diag}(\mathbb{R}_+^3), \mathbf{t}_i \in \mathbb{R}^3. \end{aligned} \quad (8)$$

Thus, we minimize difference between coordinates $g_{\mathbf{x}}$ of a cluster C_i mapped using simple translation, rotation and scaling operation to the distribution of coordinates of the style scene $g'_{\mathbf{x}}$. This constrained optimization problem can efficiently partition the problem into a collection of smaller problems of minimizing $\mathcal{SD}_{2, \gamma}^2(C_i, C'_i)$ between every content cluster C_i and assigned style cluster C'_i .

Finally, we can define a selection mapping D_i that given optimized parameters $\mathbf{t}_i, \mathbf{R}_i, \mathbf{S}_i$ selects Gaussians from the style scene $\mathcal{G}_{\text{style}}$.

$$D_i : C_i \rightarrow \mathcal{G}_{\text{style}}, D_i(C_i) = \bigcup_{g \in C_i} \mathcal{N}_k(\mathbf{S}_i \mathbf{R}_i (g_{\mathbf{x}} - \mathbf{t}_i)), \quad (9)$$

where $\mathcal{N}_k(g)$ denotes k nearest neighbors of the Gaussian g in the set of style Gaussians $\mathcal{G}_{\text{style}}$.

Ultimate stylization objective. With those preliminary steps we have successfully partitioned the problem into a collection of smaller OT problems that we can solve computing the Sinkhorn divergence between the distributions:

$$\mathcal{L}_{opt} = \sum_{i \in \{1, \dots, N\}} \mathcal{SD}_{2, \gamma}^2(C_i, D_i(C_i)). \quad (10)$$

To elucidate the computation of the Sinkhorn Divergence on the set of Gaussians, we utilize both the coordinate and brightness attributes. This alignment ensures consistency not only in coordinates but also in the overall shading of the image between content clusters and their respective style clusters. In the experimental section we visualize different results obtained optimizing color or brightness components. The optimization process outlined in Eq. (8) aims to minimize the loss over a set of parameters $\{\mathbf{t}_i, \mathbf{R}_i, \mathbf{S}_i \mid i \in 1, \dots, N\}$, while Eq. (10) optimizes over the color and coordinate components of the Gaussians $\{D_i(C_i) \mid i \in 1, \dots, N\}$.

4 Experiments

We conduct a comprehensive evaluation of WaSt-3D through a series of experiments. This includes qualitative and quantitative comparisons in Sec. 4 and further ablation studies in Sec. 4.1. For high resolution video results please refer to the supplementary materials.

Implementation Details. Our model is implemented in PyTorch. Initially, we pretrain content and style scenes using the standard Gaussian splatting training code, with additional regularization applied to the style scenes as specified in Sec. 3.2. All our experiments are conducted on a single A100 GPU, requiring 16GB of VRAM for the largest style scenes containing 8e6 Gaussians. On average our optimized scene contains 1e6 – 4e6 Gaussians. The complete optimization pipeline takes approximately 8 minutes given pretrained style and content scenes.

For Sinkhorn divergence computation, we utilize the GPU-optimized implementation available in the `geomloss` library, which relies on the `pykeops` library. Detailed algorithm is provided in the supplementary materials.

By default, we partition the content scene into 400 clusters. In Fig. 7 we show stylization results using fewer clusters. Prior to this partitioning, we sample points on the surface of the content scene to ensure that Gaussians are fitted only to the surface, mitigating issues stemming from the default Gaussian splatting fitting algorithm. This process is described in in the supplementary materials.

An additional step we need to perform is adjusting the scaling \mathbf{S} of individual Gaussians after they have been optimized with the loss defined in Eq. (10). We adjust their scale based on the changes in the average distance to the nearest neighbors. Further details are provided in the supplementary materials.

Table 1: Quantitative Comparison. WaSt-3D surpasses other methods in both CLIP high-level details similarity and human preference score. Our model also takes less time to stylize a scene. Although it requires slightly more memory, this is a reasonable trade-off considering the significant increase in stylization quality.

Method	CLIP high-level details similarity \uparrow	Human preference \uparrow	Time \downarrow	VRAM \downarrow
ARF	74.79 %	12.5%	11 min	9GB
Style-RF	74.94 %	1.5 %	18 min	6GB
SNeRF	76.99 %	10.5 %	30 min	8GB
Ours	84.40 %	75.5 %	8 min	16GB

Datasets. For our research, we need two type of 3D scenes: style and content. As content examples we used three publicly available datasets. NeRF Synthetic [32] is synthetic dataset of 3D objects with realistic non-Lambertian materials. For every scene it provides a set of 360 views and exact camera parameters. We also evaluated our dataset on real-world datasets with complex geometry LLFF [31] and Tanks&Temples [18]. Local Light Field Fusion(LLFF) is a bounded real world dataset captured with a handheld cellphone. Tank&Temples is a real world 360 scene dataset. The selected scenes exhibit diverse capture styles, including both bounded and unbounded settings. For style examples, we used scenes sourced from BlenderKit. All scenes are linked on the project page, and some are showcased in the supplementary material.

Qualitative comparison. We compare WaSt-3D with three recent stylization approaches: ARF [54], StyleRF [29] and reimplemented by Ref-NPR [55] SNeRF [33]. As illustrated in Fig. 3, our method excels in both high-quality stylization and preserving the essential features of content scenes. Demonstrating robustness, WaSt-3D outperforms other methods across diverse content scenes and varying numbers of style scenes. Other methods fail to adequately preserve style details. For instance, when applied to a *grass* style scene, only our method successfully maintains the intricate texture of the grass, showcasing the superior detail preservation achieved by our approach. More results are presented in Fig. 4, supplementary materials and on the project page.

Quantitative comparison. As demonstrated in Tab. 1, WaSt-3D performance was evaluated using the CLIP [36] similarity score. We extracted crops from the stylized scenes and compared them with crops from the original style scene to evaluate ability of each method to preserve fine details. The results presented in Table 1 demonstrate that our method outperforms other methods in this regard.

Furthermore, to evaluate the stylization quality from a human perception standpoint, we conducted a user study. Participants were tasked with selecting the most appropriate and appealing stylization for pairs of content and style scenes. As shown in Tab. 1, WaSt-3D achieved the highest scores in the user

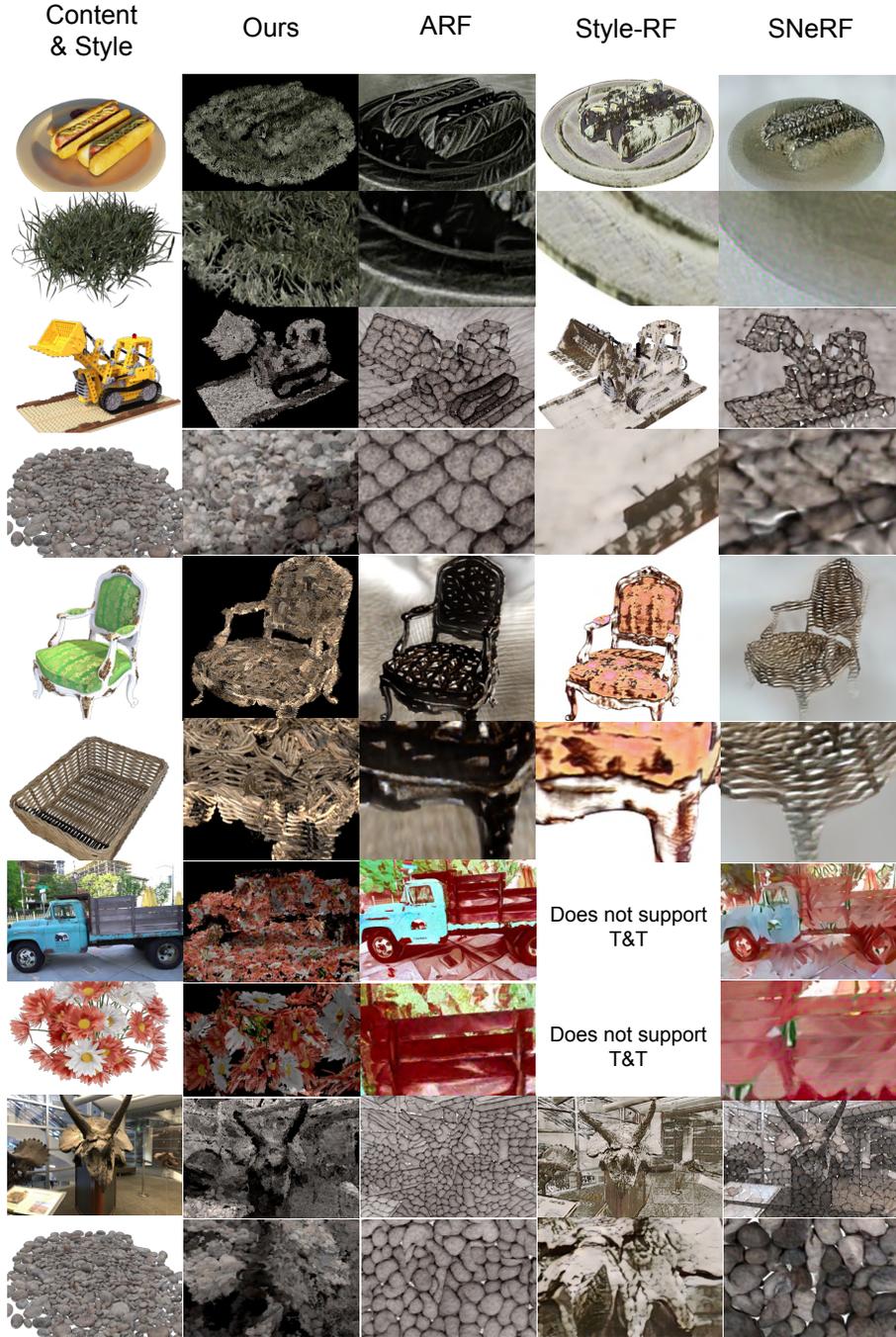


Fig. 3: Comparison of WaSt-3D against three different approaches: ARF [54], SNeRF [33], and StyleRF [29]. We conducted the comparisons on three NeRF Synthetic scenes: *hotdog*, *lego*, and *chair*; one scene, *truck*, from the Tanks&Temples dataset [18]; and the scene *horns* from the LLFF dataset [31]. Style scenes are named *grass*, *pebbles*, *wicker basket*, *bouquet*, and *stones*; examples of these style scenes are provided in the supplementary materials. For each image, we provide a close-up of the same fragment across different methods. It is important to note that while overall stylization seems comparable for small images, our method delivers favorable results for high-resolution images. Moreover, our method preserves the 3D geometry of the style scene. Please refer to the project page for video renderings of the scenes in high resolution.

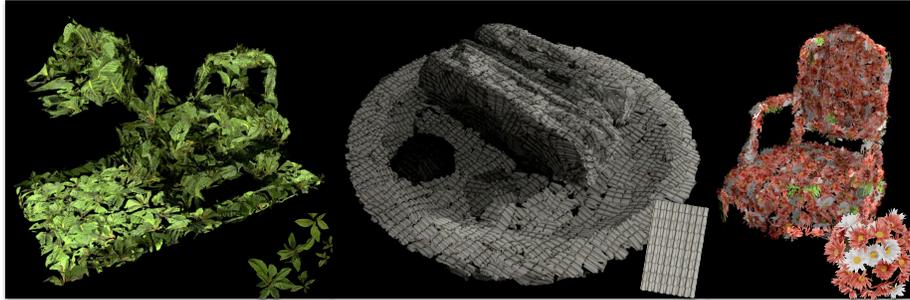


Fig. 4: Additional visual results for our method on NeRF Synthetic objects [32]. The style scene for each object is depicted in the lower right angle for optimal presentation.

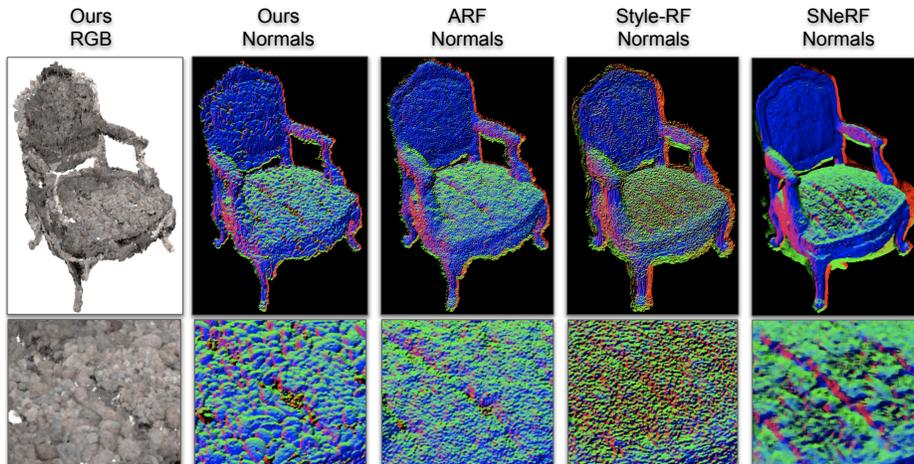


Fig. 5: Visualization of the normals obtained after stylization of the *chair* scene using the *pebbles* style with different methods: ARF [54], Style-RF [29] and SNeRF [33].

study. Additionally, we show in Tab. 1 speed and memory requirements of baselines in comparison to our method

4.1 Ablation studies

Style scene geometry preservation. To demonstrate the effectiveness of our approach in preserving the geometry of the original style scene, we compare the normals generated by our method with three alternative approaches: ARF [54], Style-RF [29] and SNeRF [33]. From Fig. 5 it is evident that our method faithfully reproduces the style geometry, while the other approaches introduce noise due to their optimization in RGB space. This highlights the robustness of our methodology in maintaining the integrity of the style scene geometry.

Sinkhorn divergence on different parameters. In Sec. 3, we outlined the computation described in Equation 10, which operates on various components of the Gaussians. Our standard configuration involves optimizing the coordinates $g_{\mathbf{x}}$ and the luminance derived from $g_{\mathbf{c}}$ using the formula $(0.299 \times R + 0.587 \times G + 0.114 \times B)$, where R , G , and B denote the values of the red, green, and blue channels within the range $[0; 1]$. As an alternative, we conducted experiments solely on the positional values $g_{\mathbf{x}}$ and on both the positional and color values $g_{\mathbf{c}}$. The outcomes of these experiments are depicted in Fig. 6. It is noteworthy that optimizing brightness (the second column) aids in preserving the original scene shading, thereby enhancing the portrayal of the scene volume.

We visualize effect of the entropy regularization term γ in the supplementary material, on default we use $\gamma = 0.05$.

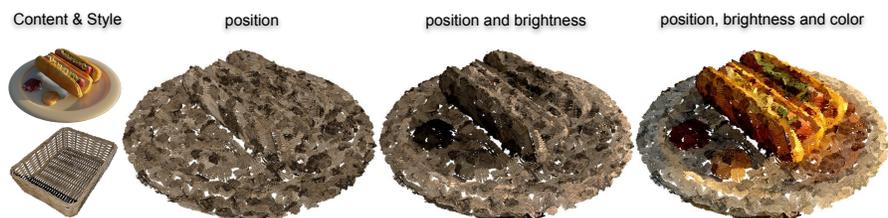


Fig. 6: Results of the stylization when minimizing \mathcal{L}_{opt} : for positions only, for position and brightness, for position, brightness and color. Original content scene *hotdog* and style scene *wicker basket* is provided in the first column.

Wasserstein distance alternatives. To prevent overstretching or tearing of the style cluster we can use the surface energy regularization. This mapping D allows translation, rotation, scaling, and stretching of individual Gaussians independently of each other. However, not controlling this operation we can mess up the original style cluster. To prevent too strong stretching of the style cluster C we compute the surface energy of every cluster before and after the transformation:

$$E(C, D(C)) := \sum_{g \in C} \sum_{h \in \mathcal{N}(g)} \left\| \|g - h\|_2 - \frac{1}{\lambda_D^2} \|D(g) - D(h)\|_2 \right\|_1. \quad (11)$$

This loss is inspired by the ARAP loss [44] that is used to model the elastic deformation of meshes, λ_D is the scaling constant of operator D . By constructing the function and surface energy we guarantee that the style will not be corrupted under any of the following transformation: translation, rotation, scaling and stretching while preserving surface energy. The results of this loss are depicted in the right column of Fig. 7. : content becomes less visible and follows the content shape poorly. We additionally replace our loss defined in Eq. (10)

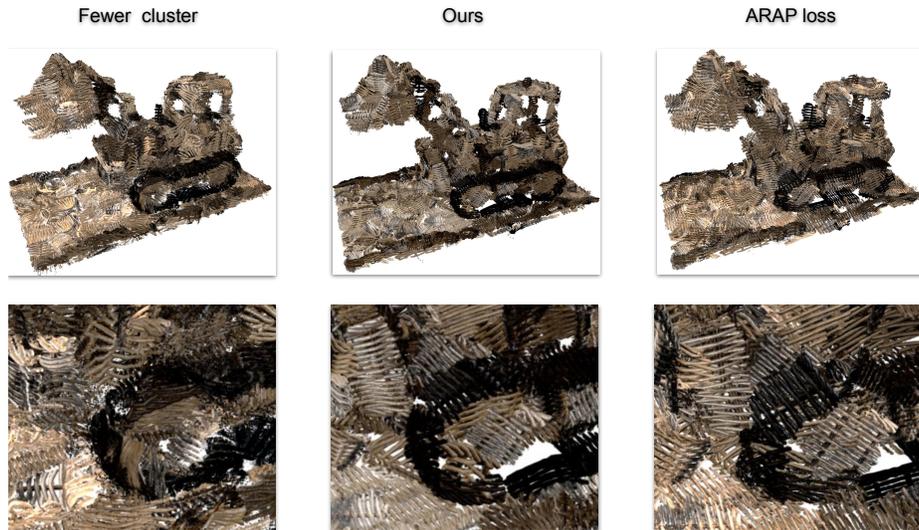


Fig. 7: We ablate two crucial components of WaSt-3D. In the middle we show our main model stylizing *lego* scene using *wicker basket* scene. On the right side we replace Sinkhorn divergence with the ARAP [44] loss. On the left side we reduce number of content cluster N from 400 to 200.

with the entropy regularized Wasserstein-2 distance Eq. (6), this makes the optimization less stable in practice and requires more hyperparameters tweaking. Results are provided in the supplementary.

5 Conclusion

In this paper, we propose the WaSt-3D method for stylizing 3D scenes using another 3D style scene as a reference. Recognizing the difficulty of general scene-to-scene translation, we first partition the content scene into simpler components, identify the best-fitting style part for each content cluster, and fine-tune the style by optimizing the Wasserstein-2 distance for each pair of clusters. This approach to matching distributions, coupled with a robust representation of the style scenes, enables a faithful reproduction of the style scene geometry. This stands in contrast to the classical approach of rendering scenes to align with feature distributions in the image encoder space, distinguishing our method from conventional 3D scene processing techniques. We conduct ablation studies on various components of our model and assess the impact of different parameters on its final performance. Additionally, we demonstrate the effectiveness of our approach through user preference studies and by measuring the CLIP similarity score of the optimized scene and the style scene patches. We believe that our approach holds promise for the community and can pave the way for alternative research directions in this field.

Acknowledgements

This project has been supported by the German Federal Ministry for Economic Affairs and Climate Action within the project “NXT GEN AI METHODS – Generative Methoden für Perzeption, Prädiktion und Planung” and the German Research Foundation (DFG) project 421703927. The authors gratefully acknowledge the Gauss Center for Supercomputing for providing compute through the NIC on JUWELS at JSC and the HPC resources supplied by the Erlangen National High Performance Computing Center (NHR@FAU funded by DFG). We also thank Ming Gui for the insightful discussion on the project and Owen Vincent for his support with technical questions.

References

1. An, J., Huang, S., Song, Y., Dou, D., Liu, W., Luo, J.: Artflow: Unbiased image style transfer via reversible neural flows. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 862–871 (2021) [4](#)
2. Arnheim, R.: Art and visual perception, a psychology of the creative eye (1967) [3](#)
3. Baatz, H., Granskog, J., Papas, M., Rousselle, F., Novák, J.: Nerf-tex: Neural reflectance field textures. Computer Graphics Forum **41** (2022) [5](#)
4. Chaudhuri, B., Sarafianos, N., Shapiro, L., Tung, T.: Semi-supervised synthesis of high-resolution editable textures for 3d humans. In: CVPR (2021) [2](#)
5. Chen, J., Ji, B., Zhang, Z., Chu, T., Zuo, Z., Zhao, L., Xing, W., Lu, D.: Testnerf: Text-driven 3d style transfer via cross-modal learning. In: International Joint Conference on Artificial Intelligence (2023) [5](#)
6. Chen, T.Q., Schmidt, M.: Fast patch-based style transfer of arbitrary style. arXiv preprint arXiv:1612.04337 (2016) [4](#)
7. Chiu, T.Y., Gurari, D.: Iterative feature transformation for fast and versatile universal style transfer. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16. pp. 169–184. Springer (2020) [2](#), [4](#)
8. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016) [2](#), [3](#), [4](#)
9. Gombrich, E.H.: The story of art (1950) [3](#)
10. Gu, S., Chen, C., Liao, J., Yuan, L.: Arbitrary style transfer with deep feature reshuffle. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8222–8231 (2018) [4](#)
11. Huang, H., Wang, H., Luo, W., Ma, L., Jiang, W., Zhu, X., Li, Z., Liu, W.: Real-time neural style transfer for videos. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 7044–7052 (2017) [2](#)
12. Huang, H.P., Tseng, H.Y., Saini, S., Singh, M., Yang, M.H.: Learning to stylize novel views. In: ICCV (2021) [5](#)
13. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE international conference on computer vision. pp. 1501–1510 (2017) [4](#)
14. Jacobs, C., Salesin, D., Oliver, N., Hertzmann, A., Curless, A.: Image analogies. In: Proceedings of Siggraph. pp. 327–340 (2001) [4](#)

15. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. ArXiv **abs/1603.08155** (2016) [2](#)
16. Jung, H., Nam, S., Sarafianos, N., Yoo, S., Sorkine-Hornung, A., Ranjan, R.: Geometry transfer for stylizing radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8565–8575 (June 2024) [5](#)
17. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (July 2023) [5](#), [6](#)
18. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics **36**(4) (2017) [10](#), [11](#)
19. Kolkin, N., Kucera, M., Paris, S., Sykora, D., Shechtman, E., Shakhnarovich, G.: Neural neighbor style transfer. arXiv e-prints pp. arXiv-2203 (2022) [4](#)
20. Kolkin, N., Salavon, J., Shakhnarovich, G.: Style transfer by relaxed optimal transport and self-similarity. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10051–10060 (2019) [2](#), [4](#)
21. Kotovenko, D., Sanakoyeu, A., Lang, S., Ommer, B.: Content and style disentanglement for artistic style transfer. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 4421–4430 (2019) [4](#)
22. Kotovenko, D., Sanakoyeu, A., Ma, P., Lang, S., Ommer, B.: A content transformation block for image style transfer. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 10024–10033 (2019) [4](#)
23. Kotovenko, D., Wright, M., Heimbrecht, A., Ommer, B.: Rethinking style transfer: From pixels to parameterized brushstrokes. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 12191–12200 (2021) [4](#)
24. Kuznetsov, A., Wang, X., Mullia, K., Luan, F., Xu, Z., Hašan, M., Ramamoorthi, R.: Rendering neural materials on curved surfaces. SIGGRAPH '22 Conference Proceedings (2022) [5](#)
25. Li, C., Wand, M.: Combining markov random fields and convolutional neural networks for image synthesis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2479–2486 (2016) [4](#)
26. Li, Y., Chen, H.y., Larionov, E., Sarafianos, N., Matusik, W., Stuyck, T.: Diffavatar: Simulation-ready garment optimization with differentiable simulation. In: CVPR. pp. 4368–4378 (2024) [5](#)
27. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Universal style transfer via feature transforms. Advances in neural information processing systems **30** (2017) [4](#)
28. Liao, J., Yao, Y., Yuan, L., Hua, G., Kang, S.B.: Visual attribute transfer through deep image analogy. arXiv preprint arXiv:1705.01088 (2017) [4](#)
29. Liu, K., Zhan, F., Chen, Y., Zhang, J., Yu, Y., Saddik, A.E., Lu, S., Xing, E.: Stylerf: Zero-shot 3d style transfer of neural radiance fields. In: CVPR (2023) [5](#), [10](#), [11](#), [12](#)
30. Mechrez, R., Talmi, I., Zelnik-Manor, L.: The contextual loss for image transformation with non-aligned data. In: Proceedings of the European conference on computer vision (ECCV). pp. 768–783 (2018) [4](#)
31. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (TOG) (2019) [10](#), [11](#)

32. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020) [5](#), [10](#), [12](#)
33. Nguyen-Phuoc, T., Liu, F., Xiao, L.: Snerf: stylized neural implicit representations for 3d scenes. ACM Transactions on Graphics **41**(4), 1–11 (Jul 2022) [5](#), [10](#), [11](#), [12](#)
34. Park, D.Y., Lee, K.H.: Arbitrary style transfer with style-attentional networks. In: proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5880–5888 (2019) [4](#)
35. Peyré, G., Cuturi, M.: Computational optimal transport. Found. Trends Mach. Learn. **11**, 355–607 (2018) [8](#)
36. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021) [10](#)
37. Ramdas, A., Trillos, N.G., Cuturi, M.: On wasserstein two-sample testing and related families of nonparametric tests. Entropy **19**, 47 (2015) [8](#)
38. Risser, E., Wilmot, P., Barnes, C.: Stable and controllable neural texture synthesis and style transfer using histogram losses. arXiv preprint arXiv:1701.08893 (2017) [4](#)
39. Sanakoyeu, A., Kotovenko, D., Lang, S., Ommer, B.: A style-aware content loss for real-time hd style transfer. ArXiv [abs/1807.10201](#) (2018) [4](#)
40. Sarafianos, N., Stuyck, T., Xiang, X., Li, Y., Popovic, J., Ranjan, R.: Garment3DGen: 3D garment stylization and texture generation. arXiv preprint arXiv:2403.18816 (2024) [5](#)
41. Segu, M., Grinvald, M., Siegwart, R.Y., Tombari, F.: 3dsnet: Unsupervised shape-to-shape 3d style transfer. ArXiv [abs/2011.13388](#) (2020) [5](#)
42. Sheng, L., Lin, Z., Shao, J., Wang, X.: Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8242–8250 (2018) [4](#)
43. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) [4](#)
44. Sorkine, O., Alexa, M.: As-rigid-as-possible surface modeling. In: Proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing. pp. 109–116 (2007) [13](#), [14](#)
45. Thonat, T., Beaune, F., Sun, X., Carr, N., Boubekeur, T.: Tessellation-free displacement mapping for ray tracing **40**(6) (dec 2021) [5](#)
46. Ulyanov, D., Lebedev, V., Vedaldi, A., Lempitsky, V.S.: Texture networks: Feed-forward synthesis of textures and stylized images. ArXiv [abs/1603.03417](#) (2016) [2](#)
47. Wang, C., Jiang, R., Chai, M., He, M., Chen, D., Liao, J.: Nerf-art: Text-driven neural radiance fields stylization. arXiv preprint arXiv:2212.08070 (2022) [5](#)
48. Wang, R., Que, G., Chen, S., Li, X., Li, J.Y., Yang, J.: Creative birds: Self-supervised single-view 3d style transfer. ArXiv [abs/2307.14127](#) (2023) [5](#)
49. Xia, X., Xue, T., Lai, W.S., Sun, Z., Chang, A., Kulis, B., Chen, J.: Real-time localized photorealistic video style transfer. 2021 IEEE Winter Conference on Applications of Computer Vision (WACV) pp. 1088–1097 (2020) [2](#)
50. Xie, T., Zong, Z., Qiu, Y., Li, X., Feng, Y., Yang, Y., Jiang, C.: Physgaussian: Physics-integrated 3d gaussians for generative dynamics. arXiv preprint arXiv:2311.12198 (2023) [6](#)
51. Xu, S., Li, L., Shen, L., Lian, Z.: Desrf: Deformable stylized radiance field. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 709–718 (June 2023) [5](#)

52. Yao, Y., Ren, J., Xie, X., Liu, W., Liu, Y.J., Wang, J.: Attention-aware multi-stroke style transfer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1467–1475 (2019) [4](#)
53. Ye, M., Danelljan, M., Yu, F., Ke, L.: Gaussian grouping: Segment and edit anything in 3d scenes. arXiv preprint arXiv:2312.00732 (2023) [5](#)
54. Zhang, K., Kolkin, N., Bi, S., Luan, F., Xu, Z., Shechtman, E., Snavely, N.: Arf: Artistic radiance fields. In: European Conference on Computer Vision. pp. 717–733. Springer (2022) [4](#), [5](#), [10](#), [11](#), [12](#)
55. Zhang, Y., He, Z., Xing, J., Yao, X., Jia, J.: Ref-npr: Reference-based non-photorealistic radiance fields for controllable scene stylization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4242–4251 (June 2023) [5](#), [10](#)