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

Dmytro Kotovenko[1], Olga Grebenkova[1], Nikolaos Sarafianos[2], Avinash Paliwal[3], Pingchuan Ma[1], Omid Poursaeed[2], Sreyas Mohan[2], Yuchen Fan[2], Yilei Li[2], Rakesh Ranjan[2], and Björn Ommer[1]

[1] CompVis @ LMU Munich and MCML
[2] Meta Reality Labs [3] Texas A&M University

1 Additional visual results

We show full size results of our method compared to the other methods in Fig. 1, Fig. 2, Fig. 3, Fig. 4, Fig. 5. Please also find high resolution video stylizations of our model on the project page.

2 Sinkhorn Iteration algorithm

We want to estimate the Wasserstein-2 distance

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

on a distributions p_s and p_c given with two sets of points N and M points respectively. We opt for optimizing entropy regularized Wasserstein-2 distance

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

since it has better convergence properties and leas to better results for our problem as shown in Sec. 3.

In this case we compute first a cost matrix of the transportation plan:

$$\begin{aligned} C = (d(x_i, y_j))_{ij} \;\; & \forall i \in \{1, N\},\;\; x_i \in p_s \\ & \forall j \in \{1, M\},\;\; x_j \in p_c \end{aligned} \tag{3}$$

and the transport plan $\pi \in \mathbb{R}_+^{N \times M}$. As a distance function $d(x, y) := \|x - y\|_2$ we use the L_2 norm.

With that computation of the Wasserstein-2 distance can be formulated as a linear optimization problem with linear constraints:

$$\begin{aligned} &\min \langle \pi, C \rangle - \gamma \sum_{ij} \pi_{ij} \log(\pi_{ij} - 1) \\ &\text{s.t. } \pi \mathbb{1}_M = p_s \\ &\quad \pi^\top \mathbb{1}_N = p_c \\ &\quad \pi_{ij} \geqslant 0 \quad \forall i \in \{1, N\}, \ \forall j \in \{1, M\}. \end{aligned} \tag{4}$$

The Lagrangian of the constrained optimization objective is:

$$\begin{aligned} \mathcal{L}(\pi, \alpha, \beta) = &\sum_{ij} (\pi_{ij} C_{ij} - \gamma \pi_{ij} \log(\pi_{ij} - 1)) + \\ &+ \alpha^\top (\pi \mathbb{1}_M - p_s) + \\ &+ \beta^\top (\pi^\top \mathbb{1}_N - p_c). \end{aligned} \tag{5}$$

To solve this we can compute the derivative of the Lagrangian with respect to the transport plan π:

$$\frac{\delta \mathcal{L}}{\delta \pi_{ij}} = C_{ij} - \gamma \log(\pi_{ij} - 1)\gamma + \pi_{ij} - \frac{1}{\pi_{ij}} \alpha_i + \beta_j. \tag{6}$$

By setting this value to zero ans simplifying the expression we find the values for the transport plan:

$$\begin{aligned} \log(\pi_{ij}) = &- \frac{1}{\gamma} (\alpha_i + C_{ij} + \beta_j) \implies \\ \pi_{ij} = &\underbrace{\exp\left(-\frac{\alpha_i}{\gamma}\right)}_{u_i} \underbrace{\exp\left(-\frac{C_{ij}}{\gamma}\right)}_{K_{ij}} \underbrace{\exp\left(-\frac{\beta_j}{\gamma}\right)}_{v_j}. \end{aligned} \tag{7}$$

Combining values u_i, K_{ij}, v_j in matrix form we obtain a new formulation for the transport plan:

$$\pi = \mathbf{diag}(\mathbf{u}) \ \mathbf{K} \ \mathbf{diag}(\mathbf{v}) \tag{8}$$

in which we know only the kernel matrix $\mathbf{K}$. Parameters $\mathbf{u}$ and $\mathbf{v}$ are unknown and depend on the Lagrange multipliers α and β respectively. By plugging in constraints $\pi \mathbb{1}_M = p_s$ and $\pi^\top \mathbb{1}_N = p_c$ from the original objective formulation 4 we obtain:

$$\begin{aligned} \pi \mathbb{1} &= \mathbf{u} \odot \mathbf{K}\mathbf{v} = p_s \\ \pi \mathbb{1}^\top &= \mathbf{v} \odot \mathbf{K}\mathbf{u} = p_c. \end{aligned} \tag{9}$$

Let's express unknown $\mathbf{u}$ and $\mathbf{v}$. By this we obtain an ultimate form of the Sinkhorn iterations:

$$\boxed{\begin{aligned}\mathbf{u} &= p_s \oslash \mathbf{K}\mathbf{v} \\ \mathbf{v} &= p_c \oslash \mathbf{K}\mathbf{u}\end{aligned}} \tag{10}$$

where $\odot$ and $\oslash$ stand for element-wise multiplication and division respectively. By alternating between computation of two terms defined in Eq. (10) it is possible to converge to some values of $\mathbf{u}$ and $\mathbf{v}$, which are used for the computation of the transport plan π as Eq. (8) suggests. For more details please see other works on computational optimal transport with applications to Wasserstein-2 distance [4, 5].

3 Entropy regularization

The Wasserstein-2 loss has one essential component, denoted as γ, which controls amount of entropy regularization. This factor plays a crucial role in smoothing the transportation plan, thereby influencing the visual results. We observe that low values of γ result in poor preservation of style scene details, while high values can aid in preserving style geometry and appearance but may cause it to deviate significantly from tightly following the content shape. By default, we utilize $\gamma = 0.07$ across all our experiments. In Fig. 6, we present two additional stylizations with low entropy regularization ($\gamma = 0.007$) and high regularization ($\gamma = 0.7$), showcasing the impact of different regularization levels.

4 Anisotropic representation.

We pointed out in the paper that our model relies on the clustering and warping procedure implemented by minimizing the Sinkhorn divergence between the style cluster and the content cluster. Since we optimize only the coordinate component $g_{\mathbf{x}}$ and the color component $g_{\mathbf{c}}$ but leave out the scaling $g_{\mathbf{S}}$ component this may result in visually unsavvory results, as Fig. 7 illustrates.

Please note that this problems stams from the original 3D gaussian splatting paper [1] which only tries to optimize for pixel similarity between the rendered image and the target image. Typically, the original training protocol results in needle-shaped gaussians as illustrated in Fig. 8.

5 Number of content clusters

Our approach consists of two three main components: splitting the content scenes into individual clusters $C = \bigcup_{i \in \{1,..,N\}} C_i$, finding best fitting cluster for each content cluster C_i using the function D_i from Eq. 10 the main script and then minimizing each C_i The number of clusters we use in our stylization directly

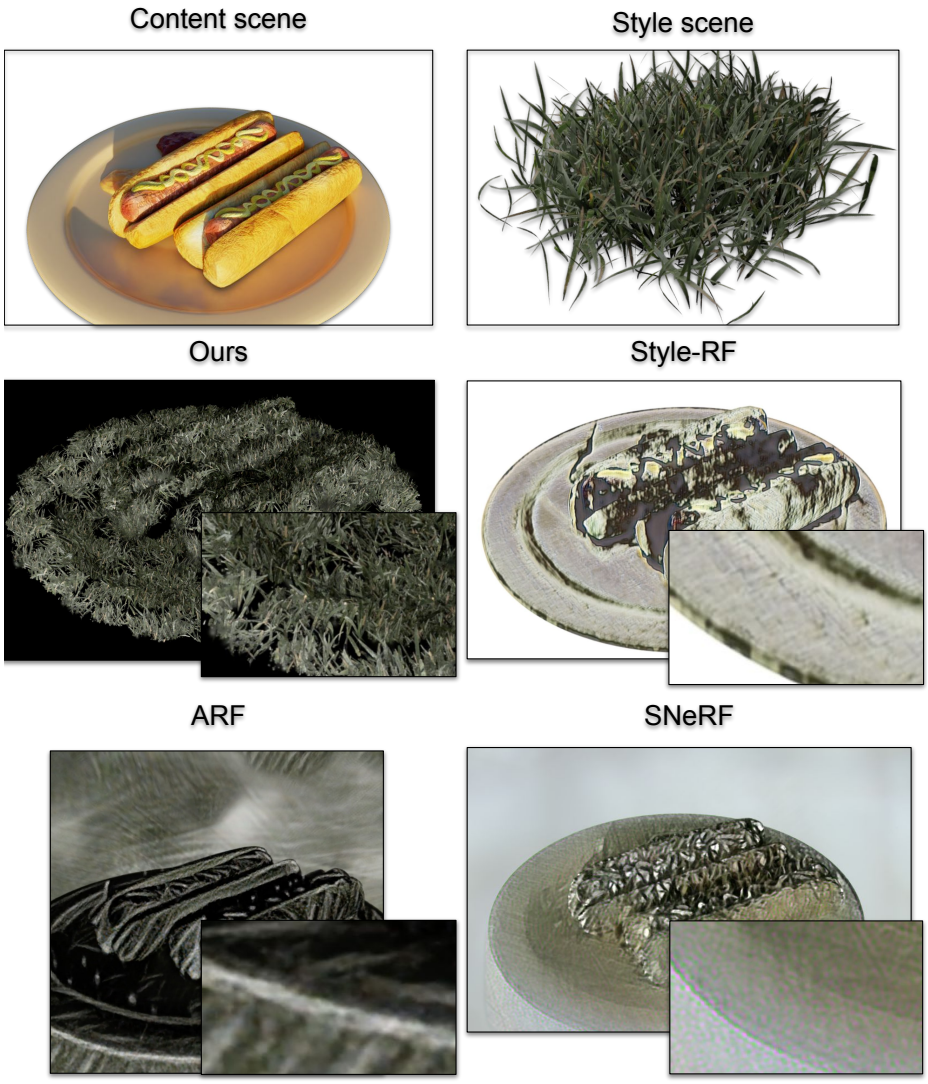


Fig. 1: Comparison of WaSt-3D against three different approaches: ARF [6], SNeRF [3], and StyleRF [2] on NeRF Synthetic scene *hotdog* and style scene *grass*.

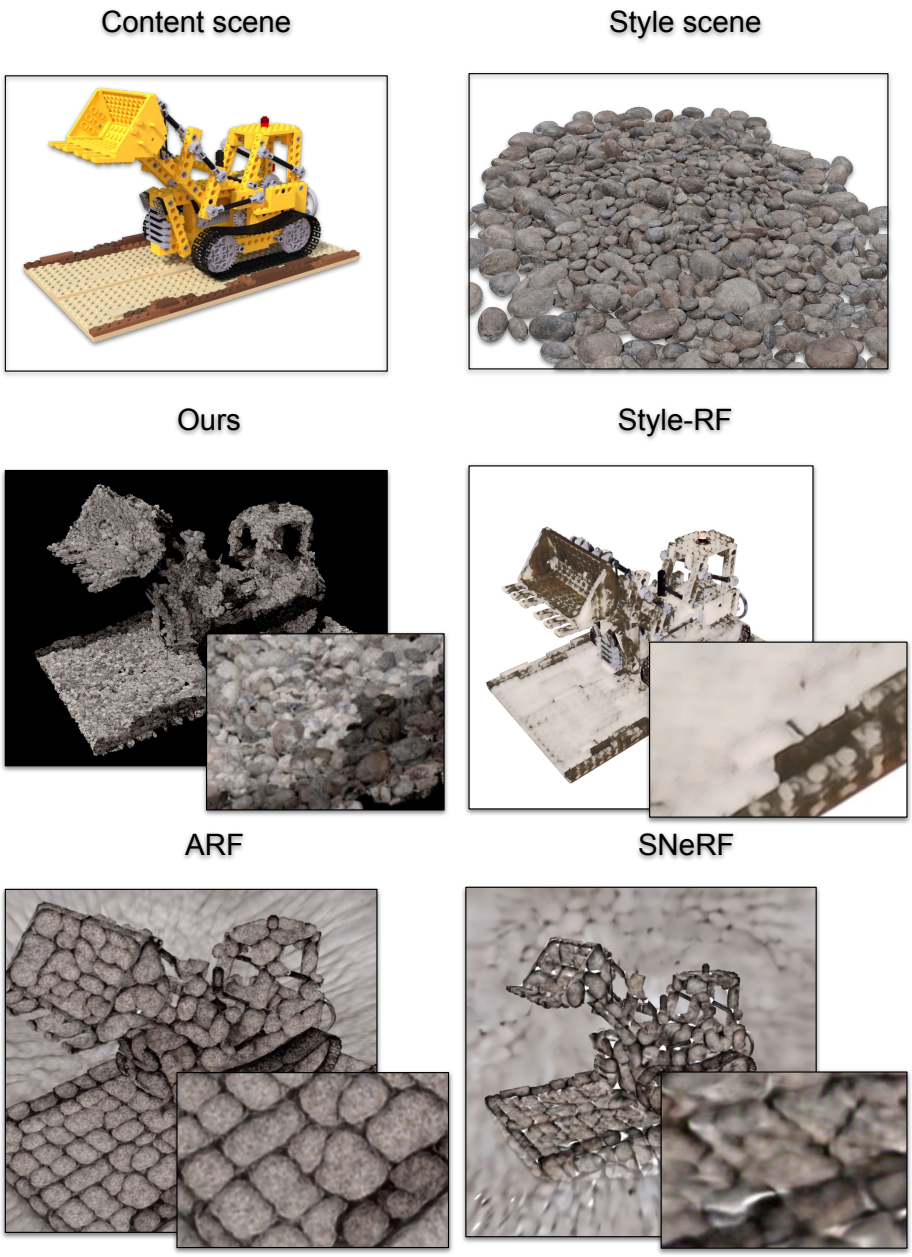


Fig. 2: Comparison of WaSt-3D against three different approaches: ARF [6], SNeRF [3], and StyleRF [2] on NeRF Synthetic scene *lego* and style scene *pebbles*.

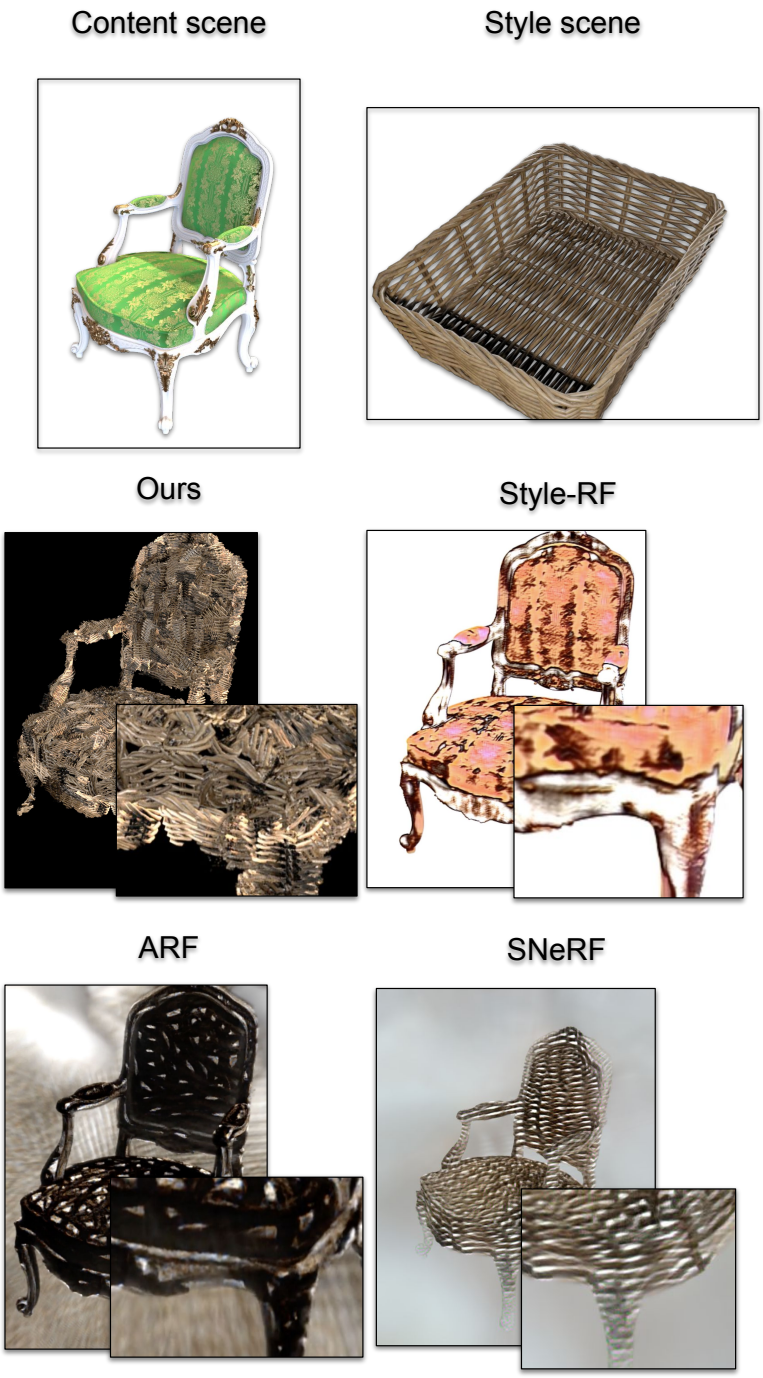


Fig. 3: Comparison of WaSt-3D against three different approaches: ARF [6], SNeRF [3], and StyleRF [2] on NeRF Synthetic scene *chair* and style scene *basket*.

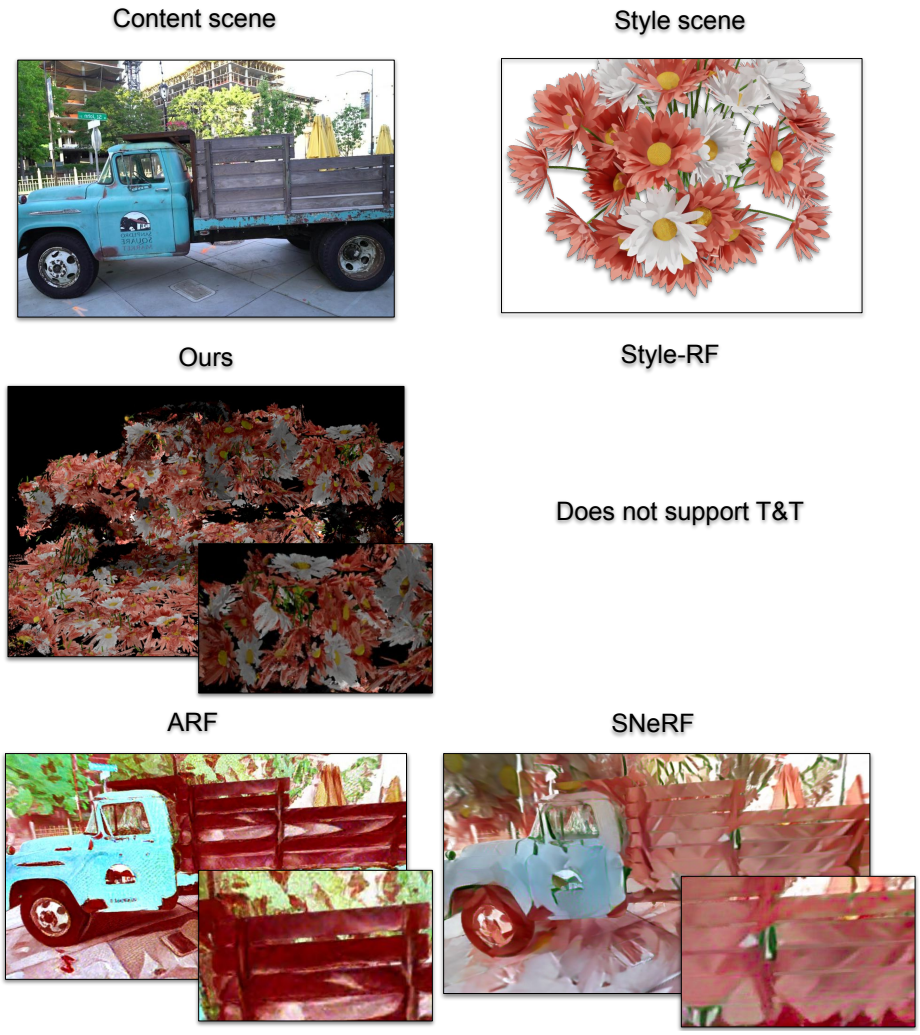


Fig. 4: Comparison of WaSt-3D against three different approaches: ARF [6], SNeRF [3], and StyleRF [2] on Tank&Temples scene *truck* and style scene *bouquet*.

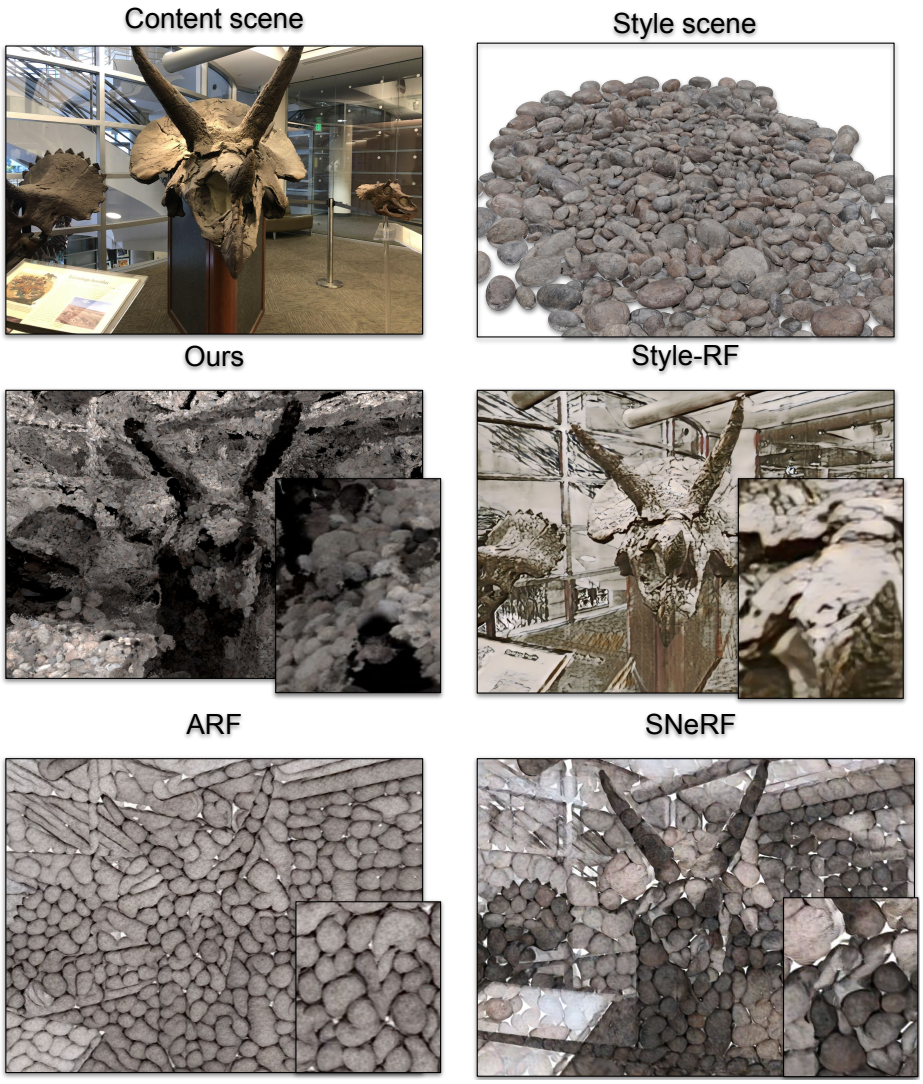


Fig. 5: Comparison of WaSt-3D against three different approaches: ARF [6], SNeRF [3], and StyleRF [2] on LLFF scene *horns* and style scene *pebbles*.

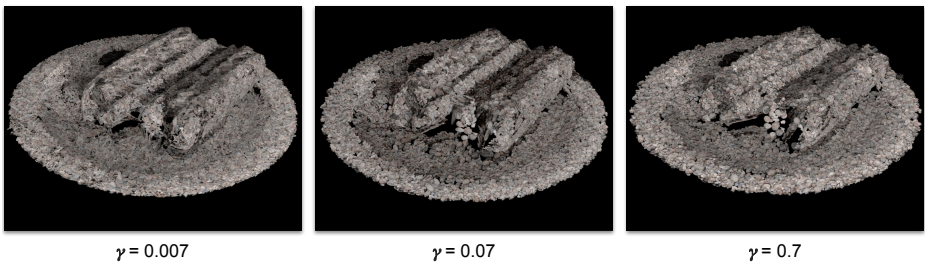
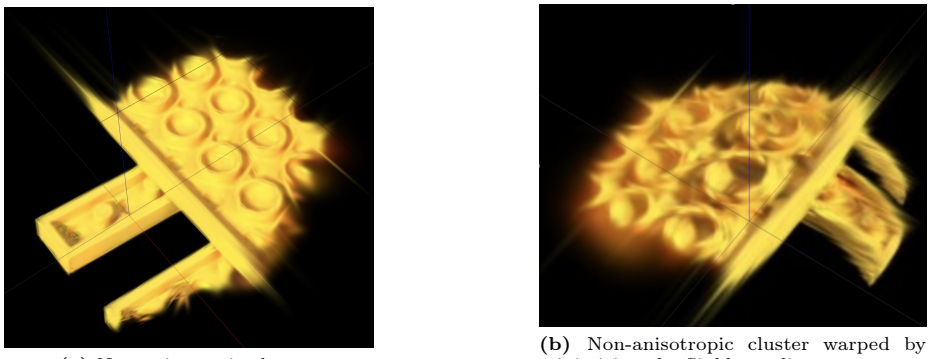


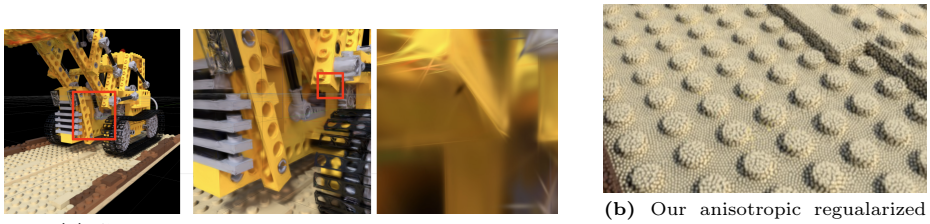
Fig. 6: Entropy component γ has clear effect on the stylization. Small regualizeation leads to poor style details. Large regularization can not follow the content shape.



(a) Non-anisotropic cluster.

(b) Non-anisotropic cluster warped by minimizing the Sinkhorn divergence.

Fig. 7: This figure highlights the importance of regularizing the Gaussian splattings representation. The left image displays a single cluster extracted from the "lego" scene of the Blender dataset. The scene is fitted using the original training pipeline outlined in the 3D Gaussian splattings paper [1]. On the right, we demonstrate that altering this representation by minimizing the Sinkhorn divergence leads to needle-shaped artifacts, which occur due to non-uniform scaling $g_{\mathbf{S}}$ of the gaussians.



(a) Cluster with non-anisotropic representation.

(b) Our anisotropic regualarized representation.

Fig. 8: Default Gaussian splattings representation is non-anisotropic. We regualarzie to get better results.

affects how detailed our stylization will be. Better clustering allows for splitting the target shape into simpler shapes that are easier to fit using function D_i. We visualize the effect the the number of clusters have on the stylization in Fig. 9, which represents how fine do we want to represent the content scene. We conduct the experiments for three different style scenes and for the number of clusters $N \in \{200, 400, 600\}$. We opt for the sweetspot of 400 cluster as our default value since it delivers visually pleasing results while not being computationally prohibitive.

6 Content scene Surface sampling

When dividing the content scene into clusters, we encounter the unexpected effect of optimizing the inner parts of the content scene as well. Consequently, during the stylization of the content scene, the inner areas are also stylized. To prevent this, we explicitly sample points on the surface of the content shape. To achieve this, we render content Gaussians $\mathcal{G}$ using the rendering function R from random viewpoints. Subsequently, we aim to extract the positions of these Gaussians using the rendered image. To accomplish this, we replace the color values $g_{\mathbf{c}}$ with the coordinate value $g_{\mathbf{x}}$. Thus, the rendered color now represents the position of the Gaussian in space. By simply sampling non-black points from the rendered image, we can determine the positions of the Gaussians. This process for a single viewpoint is illustrated in Fig. 10.

7 Sinkhorn divergence alternative

Our approach comprises three main components: first, splitting the content scenes into individual clusters $C = \underset{i \in 1,..,N}{\cup} C_i$; second, finding the best-fitting cluster for each content cluster C_i using the function D_i as described in equation Eq. 10 the main script; and finally, minimizing Sinkhorn divergence for each C_i and corresponding style cluster $D_i(C_i)$. The number of clusters directly affects the level of detail in our stylization. Better clustering enables the decomposition of the target shape into simpler shapes, making them easier to fit using function D_i. We visualize the effect of the number of clusters on the stylization in Fig. 9, representing the granularity of the content scene representation. We conduct experiments for three different style scenes and for the number of clusters $N \in 200, 400, 600$. We choose 400 clusters as our default value, as it produces visually pleasing results without imposing excessive computational demands.

8 User study background

In user study each respondent was presented with a style image, an image of a content scene, and four stylizations (ours, ARF, SNeRF, StyleRF) listed in random order. The users were asked to pick the best stylization. In total, we had eight participants in the study, some of whom had a background in different

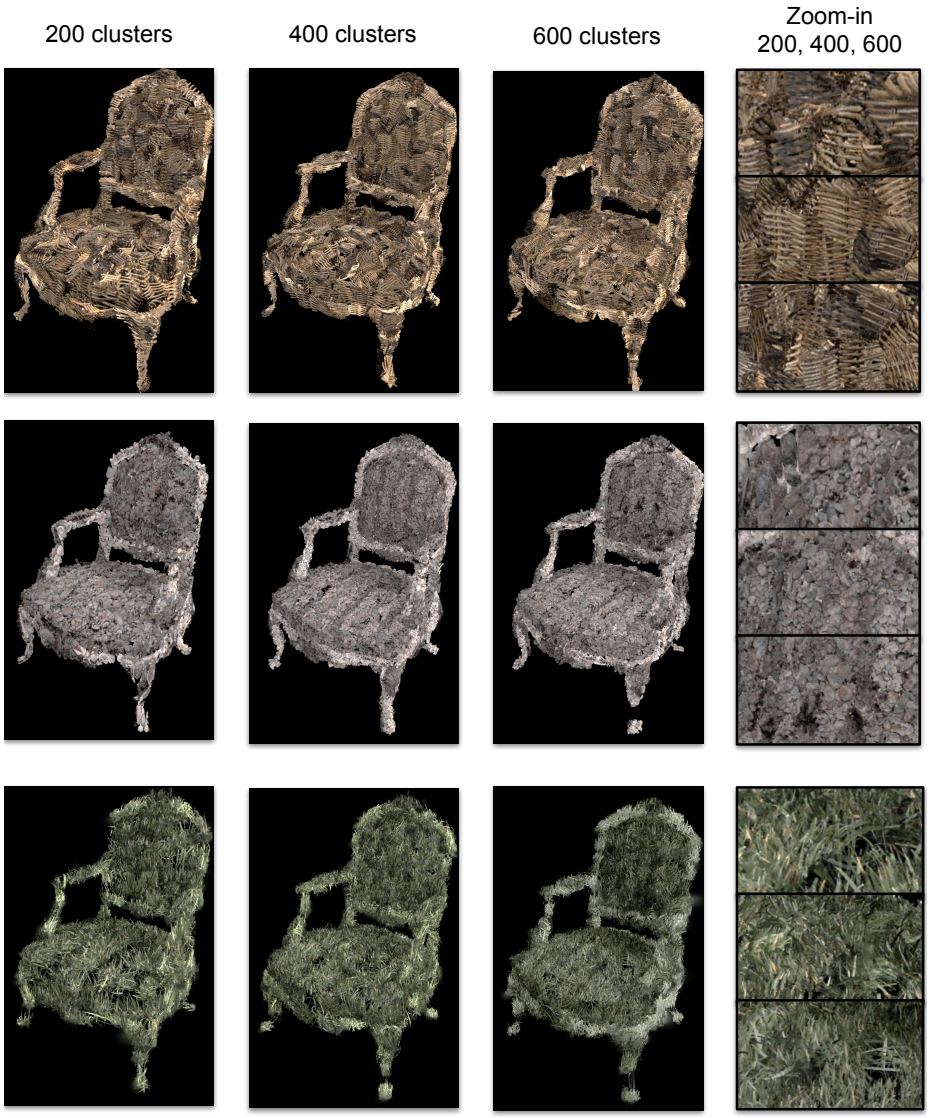


Fig. 9: we visualize performance of our model for different number of clusters: 200, 400 and 600. On the left we visualize crop from the same area from different stylizations.

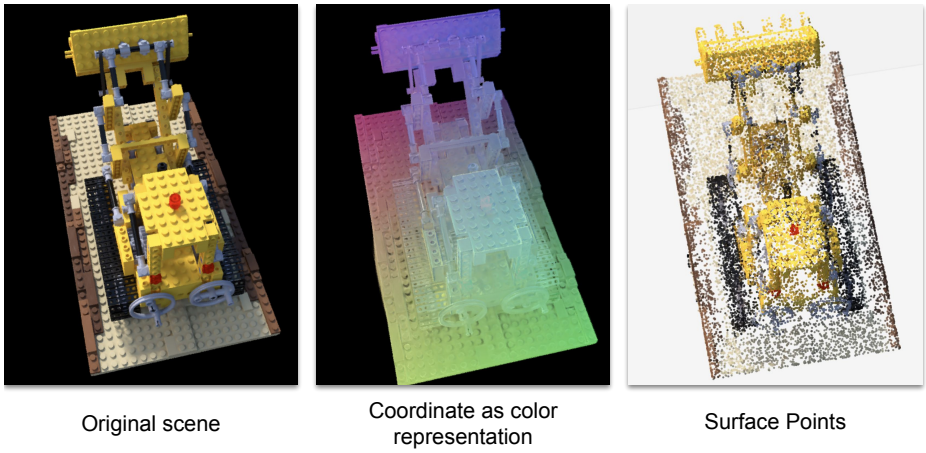


Fig. 10: We sample points on the surface of the content scene(left) by encoding coordinates in color(center) and then sampling non-black points(right).

fields of visual studies. Each participant evaluated 25 trial rounds. In each trial round, users had six seconds to make a choice.

9 Image-to-scene stylization

One of the main features of our approach is to preserve the 3D geometry of style scene. While our style examples mainly focus on complex 3D geometry, our method can also work with flat style images.

To showcase the ability of our method to extract style from the abstract artwork, we compare our model with the conventional image-to-scene stylization approach on two paintings. We turn each artwork into a plane with the painting on top, render these scenes from multiple viewpoints and represent them as 3DGS. Finally, we apply our stylization approach to these style scenes. Fig. 12 shows that in this limited scenario, our approach captures more faithfully the color distribution and artistic style of the style image resulting in more visually appealing stylized scenes.

10 Examples from style dataset

Renders from style scenes are presented in Fig. 13.

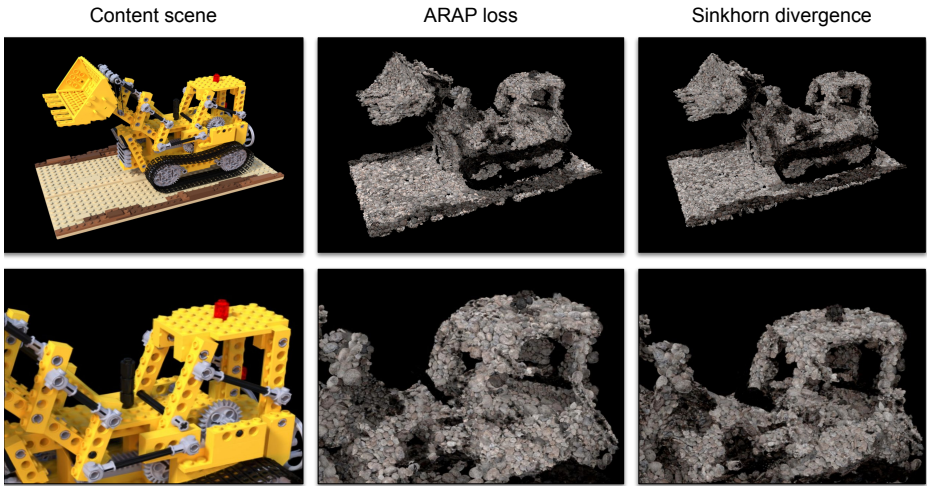


Fig. 11: In comparison with ARAP loss Sinkhorn divergence gives results more correlated with the original shape of the content scene. The effect of poor shape preservation using the ARAP loss can be easily seen on roof of *lego* scene, compared to the Sinkhorn divergence.

Fig. 12: Comparison of WaSt-3D against StyleRF on NeRF Synthetic content scenes *chair* and *hotdog* in limited style scenario.

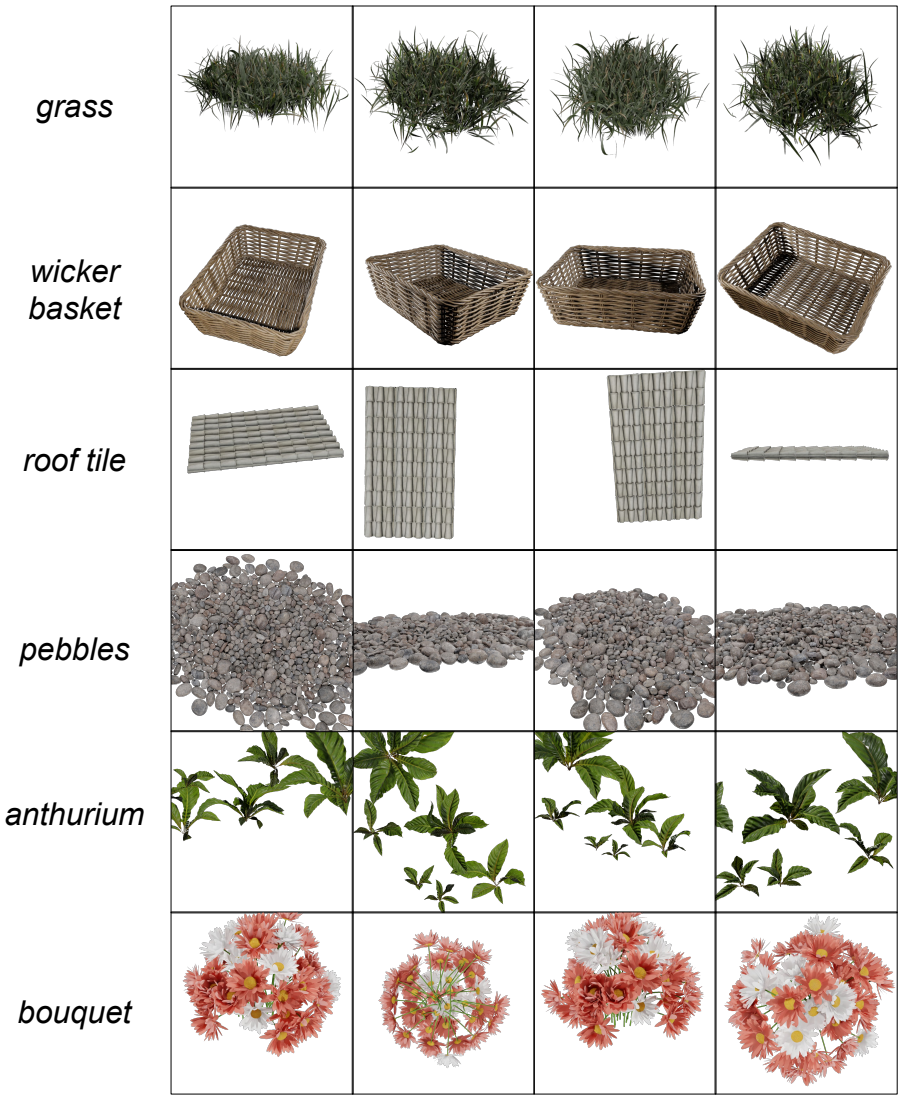


Fig. 13: Examples of style scenes used for evaluating WaSt-3D.

References

1. 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) 3, 9
2. 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) 4, 5, 6, 7, 8
3. 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) 4, 5, 6, 7, 8
4. Peyré, G., Cuturi, M.: Computational optimal transport. Found. Trends Mach. Learn. **11**, 355–607 (2018) 3
5. Solomon, J.M., de Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A.T., Du, T., Guibas, L.J.: Convolutional wasserstein distances. ACM Transactions on Graphics (TOG) **34**, 1 – 11 (2015) 3
6. 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, 6, 7, 8