GAUSSIAN SPLATTING JOINT MODELING OF GEOMETRY AND TIME-DEPENDENT RADIOMETRY

ABSTRACT

We introduce an extension to the 3D Gaussian Splatting (3DGS) framework, designed to provide novel view synthesis under varying illumination conditions throughout different times of the day. The goal of the proposed method is to enable providing a solution to the novel view synthesis problem of ”How this scene is going to look like from an unseen position and an arbitrary time of the day at an arbitrary date”. The proposed method enables the synthesis of views from unseen camera positions and at unseen times, and hence unseen illumination conditions. Traditional methods like Neural Radiance Field (NeRF) and earlier 3DGS models are typically restricted to static scenes with consistent lighting, which limits their applicability in dynamic real-world environments. We extend the 3DGS method by integrating time-dependent appearance modeling using non-causal Markovian modeling of Spherical Harmonics (SHs) and opacities, enabling rendering of 3D scenes with high fidelity and real-time performance across chronologically sampled and unsampled times.

Keywords: Novel view synthesis; 3D Gaussian splatting; Time-Dependent appearance modeling;

1. INTRODUCTION

Novel view synthesis is the challenging task of reconstructing a 3D scene from a limited set of observations, such that images of the scene can be generated from unobserved camera positions and unseen illumination conditions. One of the major challenges for novel view synthesis is the estimation and modeling of the scene's appearance and illumination at different times, sampled and unsampled.

Neural Radiance Field (NeRF) based methods [9, 1, 10, 11, 13, 3, 8] approach this task by learning a combination of a density field and a viewing-direction-dependent color field. These approaches require evaluating multiple samples from the field for each pixel to approximate the volumetric integration, which makes the methods slow in runtime aspects. On the other hand, 3D Gaussian Splatting (3DGS) [5] offers real-time rendering and high-quality rasterization results. 3DGS reconstructs the scene using a set of 3D Gaussians. Extensions of 3DGS [15, 6, 16, 2, 14] provide solutions for rendering scenes “in the wild” from samples taken at different times and using observations of the scene in different illuminations and appearances. These methods generate scene representations across various times non-chronologically, treating the different lighting conditions as distinct, isolated illuminations without considering the sequence or timing of these light changes along date and time.

NeRF, [9] and 3DGS, [5] originally addressed the problem of novel view synthesis where the geometry is fixed and illumination conditions are consistent. We propose an approach that provides an extension to the 3DGS method providing a solution for novel view synthesis under non-consistent illumination conditions while allowing the modeling of scene appearance at unsampled times. More specifically, the goal of the proposed method is to enable providing a solution to the novel view synthesis problem which can be expressed as: ”How this scene is going to look like from an unseen position and an arbitrary time of the day, at an arbitrary date”.

Therefore, the proposed approach employs the date/time-stamp part of the metadata attached to every image taken by present-day cameras in order to incorporate into the scene modeling process the different times and appearance conditions in those times to enable the synthesis of a fixed geometry 3D scene with a time-dependent appearance model. The proposed method aims to generate novel views of the scene from unsampled camera positions and at unsampled times during the day at some required date.

2. PRELIMINARIES: 3D GAUSSIAN SPLATTING

3D Gaussian Splatting (3DGS) is a method for novel view synthesis of a 3D static scene, modeling the scene by a set of 3D Gaussians. Its input is a set of observations (images) and their corresponding camera poses. This approach performs real-time rendering using a differentiable rasterizer that projects the 3D Gaussians to the 2D image plane. In this model each 3D Gaussian represents a volume of the 3D observed space such that its opacity weighted contribution at some point $p \in \mathbb{R}^3$ is given by

$$G_i(p) = \alpha_i \cdot e^{-\frac{1}{2}(p-\mu_i)^T \Sigma_i^{-1}(p-\mu_i)} \tag{1}$$

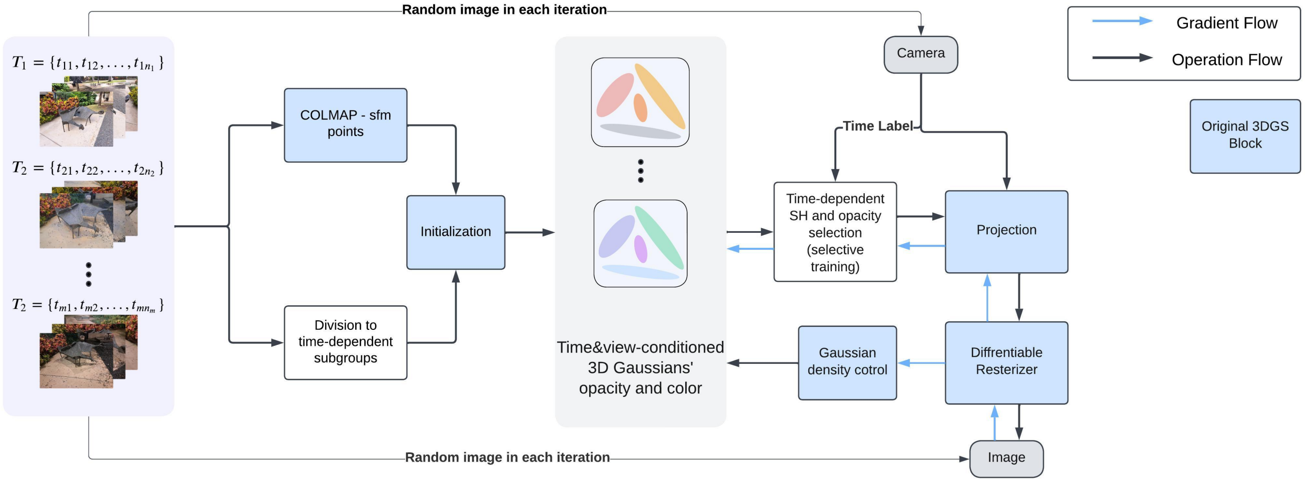

Random image in each iteration
$T_1 = \{t_{11}, t_{12}, \ldots, t_{1n_1}\}$
$T_2 = \{t_{21}, t_{22}, \ldots, t_{2n_2}\}$
$T_2 = \{t_{m1}, t_{m2}, \ldots, t_{mn_m}\}$
COLMAP - sfm points
Initialization
Division to time-dependent subgroups
Time&view-conditioned 3D Gaussians' opacity and color
Camera
Time Label
Time-dependent SH and opacity selection (selective training)
Projection
Diffrentiable Resterizer
Gaussian density cotrol
Image
Random image in each iteration
Gradient Flow
Operation Flow
Original 3DGS Block

Fig. 1: Training architecture. White boxes: our time-aware radiometry modeling blocks. Blue boxes: 3DGS blocks.

where μ_i is the center of the i-th Gaussian, and $\Sigma_i = \mathbf{R}_i \mathbf{S}_i \mathbf{S}_i^T \mathbf{R}_i^T$ its covariance matrix while $\mathbf{S}_i$ and $\mathbf{R}_i$ are the scaling and rotation matrices, respectively. α_i denotes the opacity associated with the i-th Gaussian.

In 3DGS [5] the information related to the color of the Gaussians is expressed using SH coefficients since they allow for efficiently representing view-dependent color for each Gaussian. Thus, for every Gaussian, its view-dependent color contribution c at some camera location x as viewed at viewing angle d, is evaluated using the spherical harmonic representation by

$$c(d) = \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} k_\ell^m Y_\ell^m(d) \tag{2}$$

where Y_ℓ^m are the spherical harmonics basis functions and $\{k_\ell^m\}_{m:-\ell \leq m \leq \ell, \ell:0 \leq \ell \leq \ell_{\max}}$ are the expansion coefficients while each $k_\ell^m \in \mathbb{R}^3$ is a set of 3 coefficients corresponding to the RGB components of c.

Following the splatting of the 3D Gaussians into the image plane, and denoting by $g_i(\widetilde{p})$ the result of the rasterization of (1), we have that the color of a pixel $\widetilde{p}$ in the image is given by

$$\hat{C}(\widetilde{p}) = \sum_{i=1}^{n} c_i(d) g_i(\widetilde{p}) \prod_{j=1}^{i-1} (1 - g_j(\widetilde{p})) \tag{3}$$

where $c_i(d)$ is the color contribution of the ith Gaussian, given by (2).

The rendered image is used for computing the Loss Function given by:

$$\mathcal{L}_{3DGS} = (1-\lambda)\mathcal{L}_1(\hat{C}, C) + \lambda \mathcal{L}_{\text{D-SSIM}}(\hat{C}, C) \tag{4}$$

where $\mathcal{L}_1(\hat{C}, C)$ is the L_1 photometric loss and $\mathcal{L}_{\text{D-SSIM}}(\hat{C}, C)$ is the SSIM loss.

However, since the task considered in this paper is that of novel view synthesis where the geometry is fixed and illumination conditions are time-dependent, color and opacity are now functions of time. Hence, the time-dependent color expression is given by

$$c(d;t) = \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} k_\ell^m(t) Y_\ell^m(d) \tag{5}$$

Evaluation of the time dependent projected opacity is described in Section 4.2.

3. RELATED WORK

3.1. Rendering static environment

Novel view synthesis has found many applications in recent years. In [9, 4, 5, 1] it is assumed that the 3D scene is constant in appearance and illumination. NeRF [9] is a groundbreaking technique that rapidly gained popularity for its ability to produce photorealistic 3D reconstructions, effectively capturing both the continuous geometric structure and the appearance dependency on the viewer's perspective. NeRF models a scene using an MLP that encodes the radiance field into weights for predicting the color and density of light at any point in a 3D space as a function of the viewing spatial position and viewing angle. It is implemented by projecting a ray through the volume and aggregating the color contributions, modulated by density, at a sequence of points along the ray. Although this approach is very accurate and provides impressive results, it is computationally demanding and hence not suitable for real-time applications. The main trade-off between the NeRF-based methods and the 3DGS-based methods is the time vs. storage trade-off; NeRF-based methods have much slower runtime compared to the 3DGS-based methods, but 3DGS methods require larger storage.

3.2. Rendering Varying Appearances

Based on the foregoing methods, upgraded "in-the-wild" methods have been developed to handle data acquired in non-static scenes and for modeling the 3D scene at different illuminations [8, 14, 2, 16, 6, 15]. These "in-the-wild" methods treat the different illuminations in the dataset as a discrete set of appearances and create an appearance feature vector for each image. Then, they synthesize the 3D scene and provide the ability to render it at a chosen appearance condition from the discrete set.

In [14, 2, 16, 6, 15] MLP is employed to train weights of a model that can generate the Gaussians SH or color from the image's appearance feature vectors.

WildGaussians [6] employs MLP to predict the affine transformation that maps the color of the base 3DGS to a color that matches the input image appearance. The inputs of the MLP are the Gaussians embeddings, the image embeddings, and the Gaussians view-dependent color. This approach enables the method to tailor the appearance of 3DGS models to the specific characteristics of individual images. Similarly, VastGaussian [7] employs a convolutional network to modify 3D Gaussian splatting output.

SWAG [2] requires a trainable embedding vector for each image, which is concatenated with the positional encoding of the Gaussian centers and the Gaussian SH coefficients. This combination is used to model local appearance variations and predict the Gaussian color directly from an MLP, and enables the representation of view-dependent complex details, effectively offsetting any blurring effects. WE-GS [14] calculates the residual to be added to the original SH coefficients, derived from the 3DGS process, using the Gaussian center position, the SH coefficients, and the appearance embeddings. The architecture of WE-GS incorporates multiple stages, including CNN, MLP, and U-net, to compute the appearance embeddings of images.

4. METHOD

Our proposed approach (see Fig. 1) expands the 3DGS method to handle and model the time-dependent appearance of static scenes at different dates and times of the day, enabling the rendering of the scene at unsampled observation times and from unseen camera positions. The approach we propose consists of two main steps: (1) **Gaussians's Appearance Time-Based Model** where we create SHs set for each Gaussian which correspond to the time and date when the observation samples were taken. (2) **Unobserved Appearance Estimation** where we estimate the unobserved appearance using a non-casual Markov model.

4.1. Training Time-Based Gaussian Appearances

Assuming the scene has been captured at M different time points, and thus at different illumination conditions, we model the 3D scene's appearance at each time point. Each 3D Gaussian is associated with M sets of SH coefficients and M opacity parameters. Each of the M models is optimized to model the matched scene's appearance at the corresponding time point. Each set of time-indexed SH coefficients provides a view-dependent-time-dependent color set: $\{c_i^m \mid i = 1, \dots, N,\ m = 1, \dots, M\}$, where N is the number of 3D Gaussians and m represents the appearance time index.

After creating SFM points with COLMAP [12] as in the original 3DGS method [5] we divide the training dataset into time-dependent sub-datasets and create M sets of SH coefficients. The optimization in each iteration is based on comparing the resulting rendered image with the original observation in the dataset.

On rendering, SH and opacity coefficients are selected by their time index which corresponds to that of the observation employed in the current iteration. As a result, the optimization in the iteration is applied only to the relevant SH and opacity coefficients of this time interval. Thus, on training completion, the parameters of each Gaussian are fixed in time, while the derived SH and opacity coefficients are time dependent.

4.2. Predicting Appearance at an Unsampled Time

We model the radiometric appearance variations by a non-causal Markovian model such that the SH coefficients independently obey the model where

$$
\begin{aligned}
&P(k_\ell^m(t_0) \Big| k_\ell^m(t_{-s}), \dots, k_\ell^m(t_{-1}), k_\ell^m(t_1), \dots, k_\ell^m(t_q)) \\
&= P(k_\ell^m(t_0) \Big| k_\ell^m(t_{-1}), k_\ell^m(t_1)) \\
&t_{-s} < \cdots < t_{-1} < t_0 < t_1 < \cdots < t_q \in \mathbf{R} \\
&m : -\ell \le m \le \ell, \ell : 0 \le \ell \le \ell_{\text{max}} \qquad (6)
\end{aligned}
$$

In general, over short times, the radiometric relation between sampling times is monotonic. Due to its simplicity, in the following we assume a linear dependency of the radiometric model in time. This assumption however, better suits short-time-interval appearance estimation. Given a time t_0 when the scene has not been sampled, we estimate its appearance by estimating the SH set of this time:

$$k_\ell^m(t_0) = \beta_p k_\ell^m(t_{-1}) + \beta_f k_\ell^m(t_1) \qquad (7)$$

for $m : -\ell \le m \le \ell, \ell : 0 \le \ell \le \ell_{\text{max}}$, where $k_\ell^m(t_{-1})$ is the nearest preceding in time available coefficient and $k_\ell^m(t_1)$ is the closest in time in the future. β_p and β_f are the predictor coefficients. The coefficients β_p and β_f are given by:

$$\beta_p = \frac{t_1 - t_0}{t_1 - t_{-1}}, \beta_f = \frac{t_0 - t_{-1}}{t_1 - t_{-1}} \tag{8}$$

Estimation of the opacity at an unobserved time follows a similar methodology to that applied for the SH coefficients. More specifically, the opacity of the i-th Gaussian at an unobserved time t_0 is given by:

$$\sigma^{-1}(\alpha_i(t_0)) = \beta_p \sigma^{-1}(\alpha_i(t_{-1})) + \beta_f \sigma^{-1}(\alpha_i(t_1)) \tag{9}$$

where $\sigma^{-1}(x)$ denotes the logit function. The Loss function we use is identical to that of the original 3DGS given by (4).

5. EXPERIMENTS

We constructed a new dataset, containing sequences of images such that each sequence is a set of time-labeled observations on a scene at varying times throughout the day. The dataset was captured using a standard smartphone camera. The experimental sequences are employed to evaluate the effectiveness of our proposed method in generating novel views from unseen camera positions and at unseen times of the day. Each experiment is performed on a different static scene using three different sequences taken at different times. The experiments take place at times when appearance changes of the scene are fast, such as during sunset times.

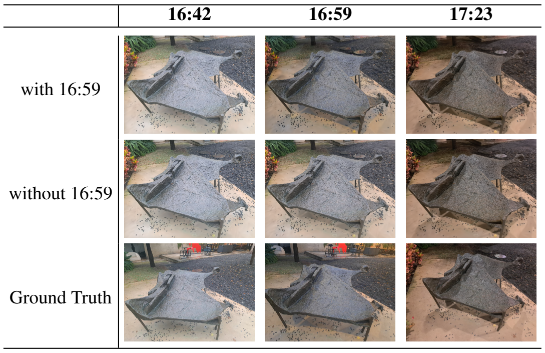


Table 1: Rendering and estimating the appearance at the 16:59 time interval on Sheep Sculpture dataset using the 16:42 interval appearance and 17:23 interval appearance. Upper row: Reconstruction of the appearance at the three time intervals with the 16:59 interval in the dataset. Middle row: Reconstruction of the appearance at the three time intervals without the 16:59 interval in the dataset. Lower row: GT appearances in the considered time intervals.

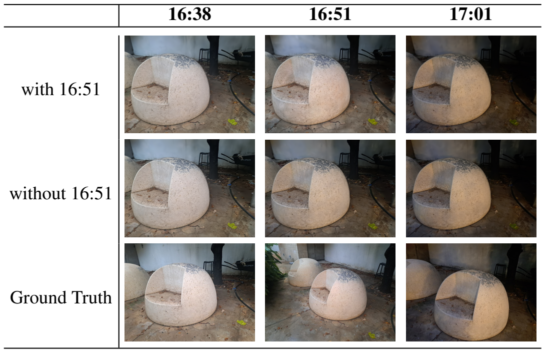


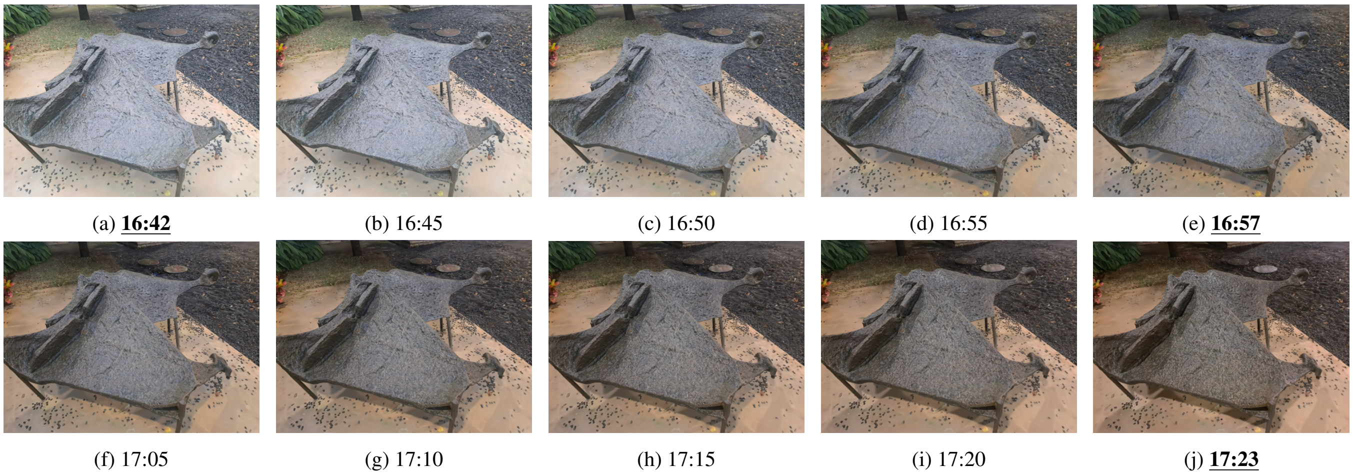
Table 2: Rendering and estimating the appearance at the 16:51 time interval on Stone-Chair dataset using the 16:38 interval appearance and 17:01 interval appearance. Upper row: Reconstruction of the appearance at the three time intervals with the 16:51 interval in the dataset. Middle row: Reconstruction of the appearance at the three time intervals without the 16:51 interval in the dataset. Lower row: GT appearances in the considered time intervals.

In Table 1 and Table 2 we present the results of rendering the scene from an unsampled camera pose and at unsampled time. Note that this evaluation method is different, and more realistic than the common practice in NeRF and 3DGS methods where evaluation is usually performed relative to images that belong to the training set - which is the test presented in the first row of Table 1 and Table 2.

Thus, the first row depicts the base experiment, *i.e.* , the result of rendering the scene from the same position but at different times where all three sequences (*i.e.* , 16:42, 16:59, and 17:23 in Table 1) are employed for training. The second row, however, presents the rendering results when the middle time interval is excluded from the training and rendering its appearance is based on the Markovian appearance model and the geometric model obtained using the data of 16:42 and 17:23, **only**. The third row provides the GT appearances from nearby positions to that of the renderings. The results demonstrate the faithful appearance reconstruction relative to the real lighting at the sampled times.

Table 3 compares the performance of our time-dependent radiometry model to that of the original 3DGS using two types of evaluations. The first evaluation examines the similarity between images rendered using the estimated 3D model and the ground truth images: The dataset contains images from two time intervals: 16:40 and 17:16, and hence with significant illumination variations. Since 3DGS assumes constant illumination conditions, the existence of illumination variations, inevitably results in performance degradation which includes appearance of ”ghost blobs” on rendering. On the other hand the proposed time-dependent appearance model does not produce such artifacts, as it is a time-aware procedure. The numerical evaluation shows the performance gain obtained using our time-dependent model over the baseline 3DGS.

Unlike the common evaluation practice for NeRF and



(a) **16:42** (b) 16:45 (c) 16:50 (d) 16:55 (e) **16:57**

(f) 17:05 (g) 17:10 (h) 17:15 (i) 17:20 (j) **17:23**

Fig. 2: Sequence of images showcasing appearance changes over time, (a), (e) and (j) are the time interval's appearances included in the training dataset; (b), (c), (d), (f), (g), (h), and (i) are estimated appearances of renderings at unobserved times. Note the faithful synthesis of the monotonicity in the radiometric changes over time.

3DGS methods, where performance evaluation is based on comparing the rendered image to an image from the training dataset, the second test involves taking a test image captured at 16:59 which is **not** available to the training procedure (in fact, the entire sequence around 16:59 is not available to the training procedure) and assessing each model ability to render the image from this specific same camera position at 16:59, based on models trained **only** with the 16:40 and 17:16 time intervals. In this test as well, the numerical evaluation demonstrates the performance gain obtained using the proposed time-dependent model over the baseline 3DGS.

Method	**Test image form training set**			**Unseen test image**		
	L1↓	**PSNR↑**	**SSIM↑**	**L1↓**	**PSNR↑**	**SSIM↑**
3DGS	0.034	26.16	0.826	0.040	25.23	0.803
Ours	0.027	27.91	0.855	0.033	26.31	0.788

Table 3: Performance evaluation of the 3D model renderings using our proposed time-dependent 3DGS and the original 3DGS.

Fig. 2 demonstrates the ability of the proposed method to track monotonic illumination variations at unsampled time points. Rendering employs training data collected at three time points only. Finally, Table 4 provides qualitative performance comparison between our method and 3DGS using the same input dataset, composed of appearances at time intervals: 16:40 and 17:27, where the goal is to estimate the appearance at 16:59. Our method estimates the appearance at 16:59, providing reconstruction with significantly fewer ghost clouds. See Supplementary for 360^o views.

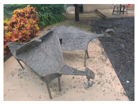
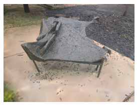
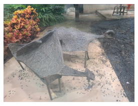
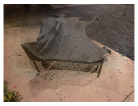
	1st point of view	**2nd point of view**
Ours 16:59 appearance estimation	 (a)	 (b)
3DGS	 (c)	 (d)
The data appearances	 (e)	 (f)

Table 4: Qualitative performance comparison between our method and 3DGS using the same input dataset, composed of appearances in two time intervals: 16:40 and 17:27. Estimating the appearance at 16:59, our method produces reconstruction with significantly fewer "ghost clouds." Unlike our method, 3DGS reconstruction shows inconsistencies in appearance when rendered from various viewpoints, (c) and (d) are different viewpoints of the same point cloud. 3DGS over-fits the input dataset, see (e) and (f) from the training dataset.

6. CONCLUSIONS

We presented an extension to the 3D Gaussian Splatting framework, augmenting it with a time-dependent appearance model. The time-dependent model employs non-causal Markovian modeling of the Spherical Harmonics and opacity parameters in the 3DGS model. It thus enables rendering 3D scenes with high fidelity and real-time performance across chronologically sampled and unsampled times. Our method enables synthesis of views from unseen camera positions and at unseen time and date, and hence unseen illumination conditions. The current model is designed to model the radiometric variations over relatively short time spans where the radiometric relation between sampling times is monotonic. Its adaptation to handle long time spans using a learned non-causal Markovian model is currently being investigated.

References

[1] Barron, Jonathan T et al. “Mip-nerf 360: Unbounded anti-aliased neural radiance fields”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 5470–5479.

[2] Dahmani, Hiba et al. “SWAG: Splatting in the Wild images with Appearance-conditioned Gaussians”. In: *arXiv preprint arXiv:2403.10427* (2024).

[3] Fridovich-Keil, Sara et al. “K-planes: Explicit radiance fields in space, time, and appearance”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 12479–12488.

[4] Fridovich-Keil, Sara et al. “Plenoxels: Radiance fields without neural networks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 5501–5510.

[5] Kerbl, Bernhard et al. “3D Gaussian Splatting for Real-Time Radiance Field Rendering.” In: *ACM Trans. Graph.* 42.4 (2023), pp. 139–1.

[6] Kulhanek, Jonas et al. “WildGaussians: 3D Gaussian Splatting in the Wild”. In: *arXiv preprint arXiv:2407.08447* (2024).

[7] Lin, Jiaqi et al. “Vastgaussian: Vast 3d gaussians for large scene reconstruction”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 5166–5175.

[8] Martin-Brualla, Ricardo et al. “Nerf in the wild: Neural radiance fields for unconstrained photo collections”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 7210–7219.

[9] Mildenhall, Ben et al. “Nerf: Representing scenes as neural radiance fields for view synthesis”. In: *Communications of the ACM* 65.1 (2021), pp. 99–106.

[10] Müller, Thomas et al. “Instant neural graphics primitives with a multiresolution hash encoding”. In: *ACM transactions on graphics (TOG)* 41.4 (2022), pp. 1–15.

[11] Rematas, Konstantinos et al. “Urban radiance fields”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12932–12942.

[12] Schonberger, Johannes L and Frahm, Jan-Michael. “Structure-from-motion revisited”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 4104–4113.

[13] Tancik, Matthew et al. “Block-nerf: Scalable large scene neural view synthesis”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 8248–8258.

[14] Wang, Yuze, Wang, Junyi, and Qi, Yue. “WE-GS: An In-the-wild Efficient 3D Gaussian Representation for Unconstrained Photo Collections”. In: *arXiv preprint arXiv:2406.02407* (2024).

[15] Xu, Jiacong, Mei, Yiqun, and Patel, Vishal M. “Wild-GS: Real-Time Novel View Synthesis from Unconstrained Photo Collections”. In: *arXiv preprint arXiv:2406.10373* (2024).

[16] Zhang, Dongbin et al. “Gaussian in the Wild: 3D Gaussian Splatting for Unconstrained Image Collections”. In: *arXiv preprint arXiv:2403.15704* (2024).