Dynamic Scene Reconstruction from Single Landscape Image Using 4D Gaussian in the Wild

In-hwan Jin*[1] Haesoo Choo*[1] Seong-Hun Jeong[2] Park Heemoon[1]
Junghwan Kim[1] Oh-joon Kwon[3] and Kyeongbo Kong[2]
*: Equal Contribution

[1] Pukyong National University
[2] Pusan National University
[3] DM Studio

Abstract. Based on the outstanding performance of 3D Gaussian splatting, recent multi-view 3D modeling studies have expanded to 4D Gaussians. By jointly learning the temporal axis with 3D Gaussians, it is possible to reconstruct more realistic and immersive 4D scenes from multi-view landscape images. However, obtaining multi-view images that accurately reflect the overall motion in the wild is extremely challenging. In the dynamic scene video field, pseudo-3D representation methods combine with Layered Depth Images (LDIs), which allows elements to render new scenes from different camera perspectives. LDIs, a simplified 3D representation of separating a single image into depth-based layers, have limitations in reconstructing complex scenes, and artifacts can occur when continuous elements like fluids are separated into layers. This paper proposes representing a complete 3D space for dynamic scene videos by modeling explicit representations, specifically 4D Gaussians, from a single image. The framework is focused on optimizing 3D Gaussians by generating multi-view images from a single image and creating 3D motion to optimize 4D Gaussians. A key aspect is consistent 3D motion estimation, which aligns common motion across multi-view images to bring 3D space motion closer to actual motions. Our model shows the ability to deliver realistic immersion in the wild landscape images through various experiments and metrics. Extensive experimental results are https://cvsp-lab.github.io/3D_MRM_page/.

Keywords: Dynamic Scene Video · 4D Gaussians

1 Introduction

The recently introduced 3D Gaussian splatting [1] provides efficient and high-quality real-time rendering by depicting scenes in three-dimensional space with multiple 3D Gaussians through an explicit representation method. This study has further extended to 4D Gaussians [2–9], incorporating a temporal dimension into 3D Gaussians to depict the dynamic scene of 3D objects' structure and appearance over time. In these research, some researchers [3, 5, 10] concentrated on modeling the variations in position, rotation, and scaling of each 3D Gaussian

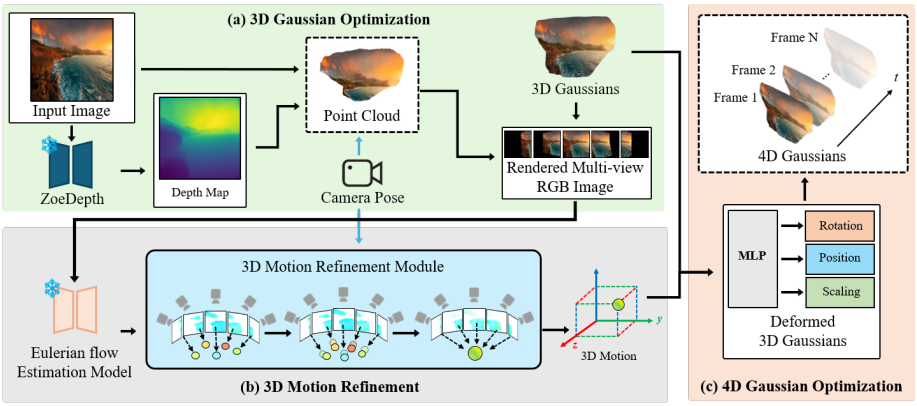

Fig. 1: The overview of our pipeline. Our framework optimizes 4D Gaussians to represent a complete 3D space, including animation, from a single landscape image in three stages.

over time, to achieve real-time dynamic scene rendering with more complex motion and efficiency.

However, real-world data, as opposed to data collected in controlled environments, is often not fully observed for reconstruction. This limitation poses challenges in achieving high-quality 3D reconstruction using existing Gaussian-based methods. Moreover, such data often fails to capture the complex motion of an entire dynamic scene, which hinders the accurate learning of temporal variations.

To address this issue, the field of creating realistic 3D videos by adding parallax to a single image without the constraints of densely captured viewpoints has been extensively studied. Among these approaches, single-image animation techniques [11–18] have been developed to infuse dynamic visual effects into static images, enabling fluid motion. Moreover, 3D photography [19–28] enables the synthesis of textures and structures in occluded areas, which facilitates the generation of parallax effects from a single image.

Recently, a field known as dynamic scene video [29, 30] has emerged, which creates videos with natural animations from large camera motion through the integration of single image animation and 3D photography. These methods employ Layered Depth Images (LDIs) [24–28, 31], which are generated by segmenting a single image into multiple layers based on depth, to represent a pseudo 3D space. However, this approach has limitations in discretely separating most elements, such as fluids, within a continuous landscape, and thus cannot fully represent 3D space. As a result, distortions can be observed, or the depth perception of the space may be reduced when the camera moves. Therefore, the achievement of complete 4D space virtualization through explicit representation, rather than relying on LDIs, is essential.

In this paper, we introduce a method to represent a complete 3D space for dynamic scene video by modeling 4D Gaussians from a single image. As illustrated in Fig. 1, the proposed framework involves of three step: (1) 3D Gaussians

Optimization, (2) Consistent 3D Motion Estimation, and (3) 4D Gaussian optimization. Initially, to optimize the 3D Gaussians, multi-view RGB images are generated from a single input image. At this stage, a 3D point cloud is created by projecting the pixels of the 2D image into a 3D space based on the estimated depth map. Next, we focus on optimizing 4D Gaussians by adjusting the regions corresponding to the motion areas of the optimized 3D Gaussians. To accomplish this, we estimate multi-view motion maps and then optimize the 3D motion within the 3D space. Finally, using the proposed 3D motion, we calculate the changes in the Gaussians' position, rotation, and scaling over time. This process allows us to optimize 4D Gaussians that maintain consistency across multiple views.

The most important part of our framework is optimization of 3D motion. Our objective is to enable movement of the 3D Gaussians, which necessitates motion within the 3D space. However, directly estimating motion in 3D space, such as with 3D point clouds or 3D Gaussians, remains largely unexplored in existing research and presents a significant challenge due to the lack of dedicated datasets for this purpose.

As an alternative, we can leverage existing off-the-shelf 2D motion estimation models [15]. Fortunately, motion estimation for 2D images has been extensively studied, which make it relatively straightforward to obtain motion information from multi-view images. Consequently, we can achieve 3D motion by lifting the estimated 2D motion into the 3D space using a depth map.

However, when the estimated motion is applied to the 3D space, we observed that motion consistency across the multi-view images is not preserved. This inconsistency occurs because the motion is estimated independently for each view. As a result, artifacts in the 4D Gaussians can arise from the movement of the Gaussians using the current 3D motion, potentially leading to distortions in the rendered dynamic scene video.

To resolve this issue, we introduce a **3D Motion Refinement Module (3D-MRM)**. The purpose of this module is to optimize 3D motion to ensure consistent motion across multi-view images. Specifically, this module initilizes arbitrary 3D motion and then refines it by minimizing the L1 loss between the projected 3D motion and the estimated 2D motion.

Our module also leverages 4D Gaussians to represent a fully 3D space with animation. Gaussian splatting provides a differentiable volumetric model that improves depth perception, enhances visual fidelity, and accelerates training times. Additionally, representing the output as Gaussians in the 3D domain provides greater versatility compared to the previous limitations of the 2D domain. These advantages enable our module to produce realistic and immersive dynamic scene videos, ensuring high visual quality even in diverse, real-world landscape images.

2 Method

In this section, we present a outline of our proposed framework, as illustrated in Fig. 1. The framework comprises three primary stages: 1) Rendering multi-view RGB images from a single image to optimize 3D Gaussians (Sec. 2.1). 2)

Estimating 3D motion to animate the 3D Gaussians. Due to inconsistencies in 3D space observed in the 2D motion maps estimated from the multi-view RGB images, we introduce a **3D Motion Refinement Module (3D-MRM)** aimed at ensuring consistent 3D motion estimation. 3) Animating the 3D Gaussians with the estimated 3D motion and optimizing the 4D Gaussians (Sec. 2.3).

2.1 Multi-view image Generation for 3D Gaussians Optimization

We use 3D Gaussians to represent a complete 3D space from a single landscape image and optimize 4D Gaussians [2–9] by incorporating motion into the represented 3D space. To achieve this, we need to generate multi-view RGB images from a single image to optimize the 3D Gaussians.

Point Cloud Generation To generate multi-view RGB images, it is necessary to convert 2D image into 3D space and project it according to various camera parameters. To achieve this, we first generate a point cloud from a single image. Specifically, we use ZoeDepth [32] to estimate the depth map $\mathbf{D} \in \mathbb{R}^{H \times W \times 1}$ for the input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$. Subsequently, we unproject from the input RGBD image $[\mathbf{I}, \mathbf{D}]$ into 3D space to generate the point cloud $\mathcal{P}$:

$$\mathcal{P} = (\mathcal{V}, \mathcal{C}) \ = \phi_{2 \to 3}([\mathbf{I}, \mathbf{D}], \mathbf{K}, \mathbf{P_0}), \tag{1}$$

where the vertices $\mathcal{V} \in \mathbb{R}^{N \times 3}$ and vertex colors $\mathcal{C} \in \mathbb{R}^{N \times 3}$ represent the point cloud of the input image. $\phi_{2 \to 3}(\cdot)$ is the function to lift pixels from the RGBD image to the point cloud, and $\mathbf{K}$ and $\mathbf{P_0}$ are the camera intrinsic matrix and the extrinsic matrix of input image $\mathbf{I}$.

Mulit-view Image Rendering Inspired by prior works [33], we project the initial point cloud onto a 2D plane image $\hat{\mathbf{I}}_i$ according to specific camera extrinsic parameters $\mathbf{P}_i$ to render multi-view RGB images as follows:

$$\hat{\mathbf{I}}_i = \phi_{3 \to 2}(\mathcal{P}, \mathbf{K}, \mathbf{P}_i). \tag{2}$$

Starting with the center camera view point $\mathbf{P_0}$, we continue the rendering process until all the cameras have been traversed. Through this process, we obtain multi-view RGB images $\hat{\mathbf{I}}_i$ of the point cloud from different angles.

Optimization with 3D Gaussian Splatting Representing the initial point cloud as a volumetric representation requires 3D Gaussians optimization. To achieve this, we initialize each Gaussian by aligning its center with the corresponding coordinate in the point cloud. Using the previously rendered multi-view RGB images, we learn a set of 3D Gaussian with all attributes to minimize the photometric loss $\mathcal{L}_{rgb}$. The photometric loss $\mathcal{L}_{rgb}$ is $\mathcal{L}_1$ combined with a D-SSIM. This process can be expressed as follows::

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

where λ is used to adjust the loss function in the optimization process. To improve the shape of the Gaussian model, continuous optimization is performed using the rendered multi-view RGB images.

2.2 Consistent 3D Motions Estimation

To animate still 3D Gaussians, we need to estimate the corresponding motion field. However, direct 3D motion generation from a point cloud or 3D Gaussians is very challenging. Fortunately, various research has been conducted in the field of single-image animation [11–18] which enable the estimation of 2D motion from a single image. Therefore, we propose a novel approach that leverages existing off-the-shelf animation models to estimate 2D motion from multi-view images. Subsequently, we unproject this 2D motion into the 3D domain to estimate 3D motion.

Multi-view 2D Motion Estimation To estimate motion maps from multi-view images at a subsequent time point, we adopt existing methods such as those by Holynski et al. [15], which estimate the motion of fluids like water and clouds from a single image. We employ an Eulerian flow $\mathbf{M}$ to simulate the motion field which indicates the movement of each pixel value, defining its velocity within the image, as follows:

$$\mathbf{F}_{t\to t+1} = \mathbf{M}(\cdot), \tag{4}$$

where $\mathbf{F}_{t\to t+1}$ represents the motion change from time t to time $t+1$, and the Eulerian flow indicates that the amount of motion change between frames moves at a constant speed and direction over time. We estimate multi-view motion maps from generated multi-view images using Holynski et al. [15]. This process can be formulated as follows:

$$\mathbf{F}_i = \mathbf{M}(\hat{\mathbf{I}}_i), \tag{5}$$

where $\hat{\mathbf{I}}_i$ represents a multi-view image. Therefore, $\mathbf{F}_i$ denotes the 2D maps of motion vectors derived from the multi-view image $\hat{\mathbf{I}}_i$.

However, since the multi-view motion maps $\mathbf{F}_i$ are estimated independently for each multi-view image, there may be a phenomenon known as ***Motion Ambiguity***. This discrepancy arises when the motion values at identical positions in 3D space do not align upon unprojection. Consequently, to guarantee uniform 3D motion estimation, we introduce a novel module called the 3D Motion Refinement Module (3D-MRM).

3D Motion Refinement Module As illustrated in Fig. 2, we use the point cloud $\mathcal{P}$, located at coordinates (x, y, z) as the starting point. Then, we define the coordinate difference between (x', y', z') and (x, y, z) as the 3D motion, $\mathbf{F_{3D}}$. This represents the movement, or motion, of the coordinate points in 3D space. Afterwards, we project the 3D motion information through camera parameters $\mathbf{K}$ and $\mathbf{P}_i$ into 2D image space, rendering it as (u_i, v_i) in the 2D space. Subsequently, the L1 loss is computed between this projected motion (u_i, v_i) and the motion map $\mathbf{F}_i$ derived from the 2D motion estimation model. Using the multi-view 2D motion $\mathbf{F}_i$ as the ground truth, we compute the loss in 2D space and optimize the 3D coordinate (x', y', z') accordingly. This process can be formulated as follows:

$$\min_{x',y',z'} \sum_{i=0}^{N} \left\| \mathbf{F}_i - \phi_{3\to 2}(\mathbf{F}_{3D}; \mathbf{K}, \mathbf{P}_i) \right\|_1, \tag{6}$$

where

$$\mathbf{F}_{3D} = (x', y', z') - (x, y, z), \tag{7}$$

N represents the total number of multi-view images generated along the camera trajectory. To minimize (6), we updated $\mathbf{F_{3D}}$ through Stochastic Gradient Descent, ultimately obtaining consistent 3D motion in space.

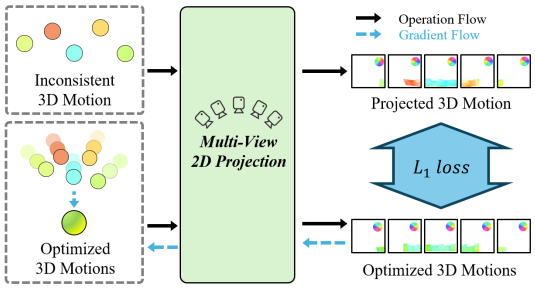

Fig. 2: 3D Motion Refinement Module. To maintain consistency of motion across multi-views, 3D motion is defined from the point cloud and projected into 2D images using camera parameters. The *L1 loss* between the projected motion and the estimated motion map as the ground truth is computed, minimizing the sum of losses for multi-view to optimize the 3D motion.

2.3 4D Gaussians Optimization

Recent studies have extended the application of 3D Gaussian models to 4D Gaussians, which allows for the representation of 3D structures and their changes over time [2–9]. Inspired by prior work [10], we focused on simply modeling the changes in position, rotation, and scaling of each 3D Gaussian over time through HexPlane and tiny Multi-Layer Perceptron (MLP).

To predict the deformation of 3D Gaussians specifically, we introduce a Spatial-Temporal Structure Encoder. This enables the effective modeling of 3D Gaussian features using a multi-resolution HexPlane and a tiny MLP. Subsequently, through a multi-head Gaussian deformation decoder, we calculate the deformation of position, rotation, and scaling ($\mathbf{\Delta x}$, $\mathbf{\Delta r}$, $\mathbf{\Delta s}$), representing each as follows:

$$(\mathbf{x}', \mathbf{r}', \mathbf{s}') = (\mathbf{x} + \mathbf{\Delta x}, \mathbf{r} + \mathbf{\Delta r}, \mathbf{s} + \mathbf{\Delta s}), \tag{8}$$

where $\mathbf{x}$, $\mathbf{r}$, $\mathbf{s}$ are initial position, rotation, and scale, respectively.

The deformations are designed to prevent distortion in the estimated 4D Gaussians. Therefore, we jointly train the first frame's multi-view images $\hat{\mathbf{I}}_i$ obtained earlier and the 2D animation results of the input image to estimate the remaining deformations. For this purpose, we draw inspiration from prior research [17, 18] and utilize a video generation model to obtain the animation. In particular, they utilize the Eulerian flow field, which represents changes in motion over time moving at a constant speed and direction, to create realistic animated looping videos for fluids, as demonstrated by Holynski et al. [15].

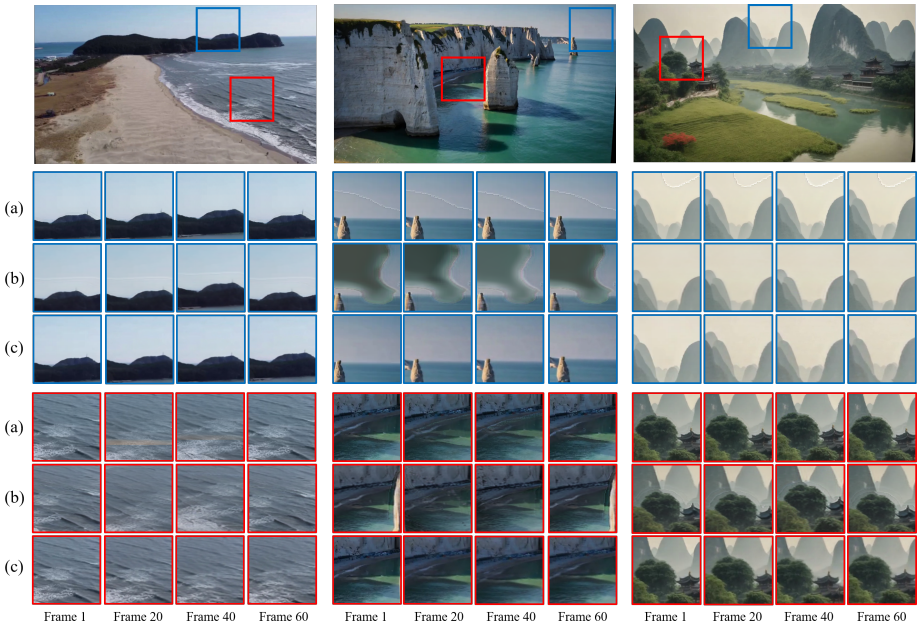

Fig. 3: Qualitative Results. (a) 3D Cinemagraphy [29], (b) Make-It-4D [30], and (c) proposed method.

3 Experiments

3.1 Experimental Setup

Baseline Model To evaluate the effectiveness of our approach in dynamic scene videos, we compared it with two state-of-the-art models: 3D-Cinemagraphy [29] and Make-It-4D [30]. Both models utilizes LDIs for 3D representation and jointly produce animations to apply the parallax effect in realistic dynamic scene video.

Unlike previous studies that limit output to 2D images, our model generates 3D Gaussian forms, which are projected to 2D based on camera trajectories for result comparison.

Implementation Details During multi-view image generation, we employ ZoeDepth [32] for depth estimation. To estimate flow from a single image we utilize Holynski et al. [15] and the pretrained single image animation model from SLR-SFS [17]. Our 3D motion optimization module is trained for about 200 iterations with a batch size 105 using the SGD Optimizer. We set the initial learning rate at 0.5 and then decaying exponentially. We conduct all experiments on a single NVIDIA GeForce RTX 3090 GPU.

3.2 Quantitative Results

Following [29], we evaluated our method and the baselines using the validation set from Holynski et al. [15]. For evaluation, we rendered the ground truth

Table 1: Quantitative Results. We compared our method's metrics against SOTA methods and applied mask to assess the animated area.

Methods	PSNR ↑	SSIM ↑	LPIPS ↓	M.PSNR ↑	M.SSIM ↑	M.LPIPS ↓
3D Cinemagraphy [29]	17.30	0.83	0.17	23.99	0.95	0.05
Make-it-4D [30]	16.98	0.81	0.20	23.75	0.95	0.05
Ours	**19.21**	**0.85**	**0.16**	**24.93**	**0.96**	**0.04**

videos from novel viewpoints using 4 different camera trajectories from 3D-Cinemagraphy [29]. In Table 1, demonstrates that our method outperforms other baselines across all metrics in view generation, which indicates that the generated views are high-fidelity and closely resemble the ground truth. Additionally, masked metrics results indicate that the animation quality in motion areas of the rendered 4D Gaussian videos was realistic and high-fidelity.

3.3 Qualitative Results

We demonstrate the superiority and high generalizability of 3D-MRM across numerous experiments. **We strongly encourage readers to view the video results on the project page at** https://cvsp-lab.github.io/3D_MRM_page/**.**

In Fig. 3, we show qualitative comparison results of our method against other baseline models, utilizing single image from real-world environments. The process of separating the input image into LDIs in 3D-Cinemagraphy [29] and Make-It-4D [30] yields artifacts in animated regions and fails to deliver natural motion, which reduces realism and visual quality.

Furthermore, due to inadequate separation between layers, objects may appear fragmented or display ghosting effects, where they seem to leave residual afterimages. In contrast, our proposed model accurately represents a complete 3D space with animations, resulting in fewer visual artifacts and enhanced rendering quality from multiple camera perspectives.

4 Conclusion

Our method efficiently obtains consistent 3D motion from multiple views by using a 2D motion estimation model without directly generating motion within the 3D space. The optimized 4D Gaussians serve as an explicit representation method, allowing users to utilize the output in the 3D domain in various ways. Additionally, our method applies well to various images, including landscape images with many objects and complex depth variations.

Our method generates consistent 3D motion from the estimated 2D motion information. However, when the 2D motion itself is unnatural, the 4D Gaussian reconstructs unnatural motion, e.g. certain regions are incorrectly recognized as static. As we are the first to generate dynamic scene videos using the volumetric representation of Gaussian splatting, we focused on generating 3D motion based on independently estimated 2D motion. Estimating complex motion in 2D will be left for future work.

References

1. Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 1
2. Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pages 800–809. IEEE, 2024. 1, 4, 6
3. Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024. 1, 4, 6
4. Richard Shaw, Jifei Song, Arthur Moreau, Michal Nazarczuk, Sibi Catley-Chandar, Helisa Dhamo, and Eduardo Perez-Pellitero. Swags: Sampling windows adaptively for dynamic 3d gaussian splatting. *arXiv preprint arXiv:2312.13308*, 2023. 1, 4, 6
5. Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4220–4230, 2024. 1, 4, 6
6. Yiqing Liang, Numair Khan, Zhengqin Li, Thu Nguyen-Phuoc, Douglas Lanman, James Tompkin, and Lei Xiao. Gaufre: Gaussian deformation fields for real-time dynamic novel view synthesis. *arXiv preprint arXiv:2312.11458*, 2023. 1, 4, 6
7. Huan Ling, Seung Wook Kim, Antonio Torralba, Sanja Fidler, and Karsten Kreis. Align your gaussians: Text-to-4d with dynamic 3d gaussians and composed diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8576–8588, 2024. 1, 4, 6
8. Yuyang Yin, Dejia Xu, Zhangyang Wang, Yao Zhao, and Yunchao Wei. 4dgen: Grounded 4d content generation with spatial-temporal consistency. *arXiv preprint arXiv:2312.17225*, 2023. 1, 4, 6
9. Jiawei Ren, Liang Pan, Jiaxiang Tang, Chi Zhang, Ang Cao, Gang Zeng, and Ziwei Liu. Dreamgaussian4d: Generative 4d gaussian splatting. *arXiv preprint arXiv:2312.17142*, 2023. 1, 4, 6
10. Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024. 1, 6
11. Yung-Yu Chuang, Dan B Goldman, Ke Colin Zheng, Brian Curless, David H Salesin, and Richard Szeliski. Animating pictures with stochastic motion textures. In *ACM SIGGRAPH 2005 Papers*, pages 853–860. 2005. 2, 5
12. Wei-Cih Jhou and Wen-Huang Cheng. Animating still landscape photographs through cloud motion creation. *IEEE Transactions on Multimedia*, 18(1):4–13, 2015. 2, 5
13. Yuki Endo, Yoshihiro Kanamori, and Shigeru Kuriyama. Animating landscape: self-supervised learning of decoupled motion and appearance for single-image video synthesis. *arXiv preprint arXiv:1910.07192*, 2019. 2, 5
14. Elizaveta Logacheva, Roman Suvorov, Oleg Khomenko, Anton Mashikhin, and Victor Lempitsky. Deeplandscape: Adversarial modeling of landscape videos. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 256–272. Springer, 2020. 2, 5

15. Aleksander Holynski, Brian L Curless, Steven M Seitz, and Richard Szeliski. Animating pictures with eulerian motion fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5810–5819, 2021. 2, 3, 5, 6, 7
16. Aniruddha Mahapatra and Kuldeep Kulkarni. Controllable animation of fluid elements in still images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3667–3676, 2022. 2, 5
17. Siming Fan, Jingtan Piao, Chen Qian, Hongsheng Li, and Kwan-Yee Lin. Simulating fluids in real-world still images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15922–15931, 2023. 2, 5, 6, 7
18. Aniruddha Mahapatra, Aliaksandr Siarohin, Hsin-Ying Lee, Sergey Tulyakov, and Jun-Yan Zhu. Text-guided synthesis of eulerian cinemagraphs. *ACM Transactions on Graphics (TOG)*, 42(6):1–13, 2023. 2, 5, 6
19. Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2
20. Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7467–7477, 2020. 2
21. Joachim Weickert. Coherence-enhancing diffusion filtering. *International journal of computer vision*, 31:111–127, 1999. 2
22. Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 551–560, 2020. 2
23. Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3d ken burns effect from a single image. *ACM Transactions on Graphics (ToG)*, 38(6):1–15, 2019. 2
24. Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8028–8038, 2020. 2
25. Johannes Kopf, Suhib Alsisan, Francis Ge, Yangming Chong, Kevin Matzen, Ocean Quigley, Josh Patterson, Jossie Tirado, Shu Wu, and Michael F Cohen. Practical 3d photography. In *Proceedings of CVPR Workshops*, volume 1, page 2, 2019. 2
26. Johannes Kopf, Kevin Matzen, Suhib Alsisan, Ocean Quigley, Francis Ge, Yangming Chong, Josh Patterson, Jan-Michael Frahm, Shu Wu, Matthew Yu, et al. One shot 3d photography. *ACM Transactions on Graphics (TOG)*, 39(4):76–1, 2020. 2
27. Qianqian Wang, Zhengqi Li, David Salesin, Noah Snavely, Brian Curless, and Janne Kontkanen. 3d moments from near-duplicate photos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3906–3915, 2022. 2
28. Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 302–317, 2018. 2
29. Xingyi Li, Zhiguo Cao, Huiqiang Sun, Jianming Zhang, Ke Xian, and Guosheng Lin. 3d cinemagraphy from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4595–4605, 2023. 2, 7, 8
30. Liao Shen, Xingyi Li, Huiqiang Sun, Juewen Peng, Ke Xian, Zhiguo Cao, and Guosheng Lin. Make-it-4d: Synthesizing a consistent long-term dynamic scene

video from a single image. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8167–8175, 2023. 2, 7, 8

31. Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 231–242, 1998. 2
32. Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 4, 7
33. Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384*, 2023. 4