Make-It-4D: Synthesizing a Consistent Long-Term Dynamic Scene Video from a Single Image

Liao Shen
leoshen@hust.edu.cn
School of AIA, Huazhong University of Science and Technology

Xingyi Li
xingyi_li@hust.edu.cn
School of AIA, Huazhong University of Science and Technology
S-Lab, Nanyang Technological University

Huiqiang Sun
shq1031@hust.edu.cn
School of AIA, Huazhong University of Science and Technology

Juewen Peng
juewenpeng@hust.edu.cn
School of AIA, Huazhong University of Science and Technology

Ke Xian*
ke.xian@ntu.edu.sg
S-Lab, Nanyang Technological University

Zhiguo Cao
zgcao@hust.edu.cn
School of AIA, Huazhong University of Science and Technology

Guosheng Lin
gslin@ntu.edu.sg
S-Lab, Nanyang Technological University

ABSTRACT

We study the problem of synthesizing a long-term dynamic video from only a single image. This is challenging since it requires consistent visual content movements given large camera motions. Existing methods either hallucinate inconsistent perpetual views or struggle with long camera trajectories. To address these issues, it is essential to estimate the underlying 4D (including 3D geometry and scene motion) and fill in the occluded regions. To this end, we present **Make-It-4D**, a novel method that can generate a consistent long-term dynamic video from a single image. On the one hand, we utilize layered depth images (LDIs) to represent a scene, and they are then unprojected to form a feature point cloud. To animate the visual content, the feature point cloud is displaced based on the scene flow derived from motion estimation and the corresponding camera pose. Such 4D representation enables our method to maintain the global consistency of the generated dynamic video. On the other hand, we fill in the occluded regions by using a pre-trained diffusion model to inpaint and outpaint the input image. This enables our method to work under large camera motions. Benefiting from our design, our method can be training-free which saves a significant amount of training time. Experimental results demonstrate the effectiveness of our approach, which showcases compelling rendering results.

*Corresponding author.

MM '23, October 29–November 3, 2023, Ottawa, ON, Canada.

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3612033

CCS CONCEPTS

• **Computing methodologies** → Computer graphics; *Image manipulation*; **Image-based rendering**;

KEYWORDS

Long-term dynamic video synthesis; Global consistency; 4D representation; Inpainting and outpainting

ACM Reference Format:
Liao Shen, Xingyi Li, Huiqiang Sun, Juewen Peng, Ke Xian, Zhiguo Cao, and Guosheng Lin. 2023. 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 (MM '23), October 29–November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3581783.3612033

1 INTRODUCTION

The ubiquity of mobile cameras has made it easy for us to capture numerous photos of beautiful landscapes during our vacations. However, these static photographs only capture a single moment, lacking temporal information and failing to convey the depth and dimensionality of the scene. To create a more engaging visual experience, it is promising to explore short-form videos that allow viewers to navigate through the scene. Given a static image as in Fig.1, humans can imagine a dynamic scene and mentally traverse through the captured space. To achieve a similar goal, in this work, we aim to synthesize a consistent long-term dynamic video that contains both visual content movements and large camera motions. This provides an immersive and realistic experience for viewers.

Nowadays, Text-to-Video (T2V) methods [10, 11, 14, 21, 36] based on the Diffusion model can generate videos from text prompts. However, they cannot control the camera pose to generate dynamic new viewpoint images and commonly require more complex diffusion model architectures and larger scale datasets. In contrast, our

Figure 1: Given a single still image, we aim to synthesize a video featuring large camera motions and visually dynamic elements like the moving clouds and the rolling sea. Existing methods either generate inconsistent novel views [18] or degrade significantly under large camera motions [16]. In contrast, our method can generate a consistent long-term dynamic video. We encourage readers to view with Adobe Acrobat or KDE Okular.

method constructs a 3D underlying representation of the scene, allowing the generated video to be controlled by camera trajectories.

In addition, some previous studies have focused on either synthesizing landscape flythroughs or animating a single image. To generate long camera track flythroughs from a single image, previous methods [1, 18, 19] continuously synthesize novel frames starting from the input image with three steps per round: render, refine, and repeat. Specifically, they intuitively perform a per-frame refinement process to inpaint the missing regions caused by small camera motions and then repeat this process to obtain a complete camera trajectory video. Such a mechanism of using network outputs as new inputs is also prone to error accumulation, ultimately resulting in domain drifting and poor output quality. Besides, these methods cannot ensure the 3D consistency of the scene due to the lack of an underlying representation of the scene. Recently, Chai et al. [2] propose a method to construct an unbounded 3D world with a persistent scene representation. Yet, the resolution of the generated video is severely limited due to the high cost of volume rendering, and its 3D consistency still cannot be guaranteed. Furthermore, it is worth noting that all flythrough methods do not consider the motion of dynamic elements in the scene, such as clouds, smoke, and water. These limitations mentioned above may result in an unrealistic generated video. While single-image animation methods [7, 12] have shown promising results in producing realistic animated videos from a single image, they often face difficulties in handling camera motion. On the other hand, 3D Cinemagraphy [16] allows for animation with slight camera motions, but it is less suitable for generating long camera trajectories. In summary, these methods suffer from the following issues: (i) perpetual view generation methods [1, 18, 19] cannot achieve global consistency since there is no explicit 3D representation; (ii) they also do not take into account the motion of dynamic elements; (iii) while 3D Cinemagraphy [16] produces plausible animation of the scene, it struggles with long camera trajectories.

To address the above challenges, we present **Make-It-4D**, a novel training-free framework that generates a long-term dynamic video from a single image. To ensure the availability of complementary content when the camera moves beyond the boundaries of the initial view, we employ a pre-trained inpainting diffusion model [30] to outpaint the input image. To ensure 3D consistency, it is sufficient to perform a scene representation of the outpainted image, without constructing the entire 3D world. Considering this, we represent the scene as layered depth images (LDIs). We then utilize the inpainting diffusion model to seamlessly inpaint occluded regions of each color layer, resulting in a realistic appearance. By doing so, we eliminate the need to fill in the little holes that appear every time the camera moves, because our method has already pre-filled the information for each layer. By doing so, our method can work under large camera motions and avoid repeatedly feeding the network's output as input. This will mitigate domain drifting and inconsistency.

To animate the scene, we use the scene flow obtained by motion estimation to animate the point cloud. Since our method involves large camera motions, we interpolate the target camera pose to obtain intermediate camera poses for the frames in between. We then use these interpolated poses to render the intermediate frames, resulting in a coherent and smooth output video. Moreover, our framework allows for optional guidance through the use of text prompts, which can be leveraged due to the advantage of using pre-trained diffusion models. Throughout the entire process, our framework brings still images to life, providing a vivid fly-through experience.

Extensive experiments demonstrate that our approach significantly outperforms the state-of-the-art landscape flythroughs synthesizing and 3D animation methods. We also conduct a user study to evaluate the performance of our method. To summarize, our key contributions include:

- We propose **Make-It-4D**, a novel method that can synthesize a dynamic video from a single image. The generated video involves both visual content movements and large camera motions, bringing the still image back to life.
- We estimate the underlying 4D of the scene, including 3D scene representation and scene motion, to ensure the consistency of the generated video. To further allow large camera motions, we use a pre-trained diffusion model to inpaint and outpaint the input image to fill in the occluded regions.
- Our framework is entirely training-free, enabling the synthesis of long-term novel views in diverse in-the-wild scenes without the need for large-scale training.

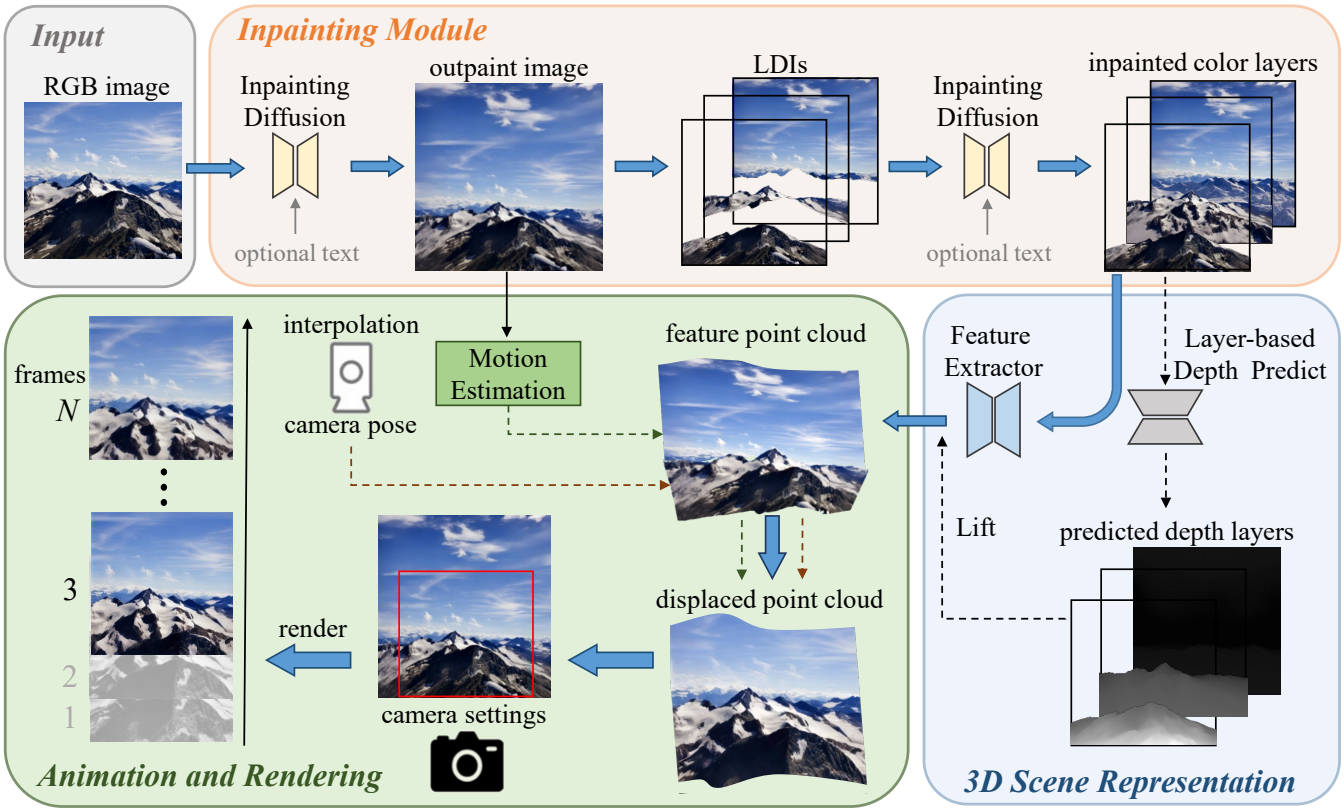

Figure 2: Overview of our method. To begin, we employ a pre-trained inpainting diffusion model to outpaint a given RGB image. Next, we proceed to predict a dense depth map of the outpainted image and utilize depth discontinuities to partition the outpainted image into layered depth images (LDIs). Subsequently, we use the inpainting diffusion model to fill in obscured regions of each color layer that are obstructed by prior layers. Following this, we perform layer-based depth prediction to obtain predicted depth layers. We then utilize a 2D feature extractor to encode features from the inpainted color layers and lift them into a 3D point cloud via their corresponding predicted depth layers. The feature point cloud is displaced based on the scene flow derived from motion estimation and the corresponding camera pose. It is important to note that we set the camera intrinsics to ensure that the view size of the feature point cloud, when projected onto the target image plane, is consistent with the resolution of the initial input RGB image. To generate intermediate frames, we interpolate between the target camera pose and the initial camera pose.

2 RELATED WORK

Long-range view synthesis from a single image. Recent methods [1, 15, 18, 19, 27, 29, 39] have proposed to synthesize scenes given a single image and camera motion as input. InfNat [19], InfNat-Zero [18], and DiffDreamer [1] utilize iterative training protocols to synthesize a video depicting an in-the-wild scene captured from long camera trajectories. However, they adopt a per-frame generation framework and lack underlying scene representations, resulting in domain drifting and inconsistent novel views. Other works [2, 4] propose to generate long-term views by building unbounded 3D worlds. This enables 3D consistent view generation but requires expensive computational costs. By contrast, our method only performs scene representations on the input image, which already inherently incorporates 3D consistency. Moreover, we can animate the dynamic objects in the scene such as clouds and lakes while the camera moves to generate novel views, which is infeasible for all previous methods. Last but not least, these methods typically require large-scale training on videos or images from the target domain, whereas our approach is entirely training-free. By leveraging a pre-trained inpainting diffusion model, our method can generate novel views from a single image across diverse categories of scenes.

Single-image animation. Single-image animation is the task of converting a still image into an animated video. Some works [5, 13] focus on animating certain objects through physical simulations, but may not be easily applicable to the more general case of in-the-wild photos. Given videos as guidance, there are many methods that attempt to perform motion transfer on static objects [3, 6, 20, 28, 34, 35]. They require a reference video to drive the motion of static objects and thus are not suitable for our task. Therefore, we focus more on methods [7, 8, 12, 16, 17, 22] that convert still images into animated video textures by exploiting motion priors learned from a single image. For example, 3D Cinemagraphy [16] jointly learns image animation and novel view synthesis in 3D. However, it struggles with large camera motions. In contrast, our method

can animate dynamic objects in the image even when the camera moves over long distances.

Text-to-3D generation. With the success of text-to-image generative models in recent years, text-to-3D generation has also gained a surge of interest in the community. DreamFusion [25] showcased impressive capability in text-to-3D synthesis by utilizing a powerful pre-trained text-to-image diffusion model [31] as a strong image prior. Text-to-Video (T2V) methods [10, 11, 36] based on diffusion model attempt to generate realistic videos from text or images. However, these methods cannot allow arbitrary camera motion. Related to our work, SceneScape [9] utilizes a pre-trained text-to-image model to generate long videos of indoor scenes based on the input text and camera poses, but can only synthesize zoom-out trajectories and is limited in its ability to handle outdoor scenes. Instead, our method can work well in outdoor scenes and allows for arbitrary camera movement. Additionally, SceneScape requires fine-tuning of its model, while our approach is training-free.

3 METHOD

3.1 Overview

To the best of our knowledge, we are the first to generate consistent 3D camera trajectories that capture the experience of flying into or out of a given RGB image, while accounting for dynamic elements in the scene such as moving clouds. At the core of Make-It-4D is a training-free framework comprising (i) an explicit 3D representation that allows our method to fly around the input image in a consistent manner, (ii) a motion estimation module that can animate the scene, and (iii) an inpainting module that enables a larger camera motion and long camera trajectories.

We schematically illustrate our pipeline in Fig. 2. Our method starts by outpainting the input RGB image using a pre-trained inpainting diffusion model [30] (Sec. 3.2). After that, we employ a pre-trained monocular depth estimator [26] to predict the depth map of the outpainted image and leverage depth discontinuities to partition the outpainted image into layered depth images (LDIs) [24, 32]. We then use the diffusion model to fill in the occluded regions of each color layer. After filling in the obscured regions, we represent the scene in 3D space using layer-based depth prediction to obtain the predicted depth layers (Sec. 3.3). Next, we utilize a 2D feature extractor to encode features from the inpainted color layers and lift them into 3D using their corresponding depth layers, resulting in a feature point cloud. The feature point cloud is then displaced based on the scene flow derived from motion estimation and the corresponding camera pose. We adjust the camera intrinsics to ensure that the feature point cloud, when projected onto the target image plane, matches the resolution of the initial input RGB image. Finally, we render the intermediate frames to generate a coherent video (Sec. 3.4).

3.2 Inpainting Module

The information obtained from a single input image is limited. To synthesize a fly-through video, we need to obtain more information about the emerging regions outside the input field of view and the obscured regions in the input image. 3D Cinemagraphy [16] appears blank regions when the camera moves out of the initial view boundaries. Besides, they use a CNN-based inpainting model from 3D Photos [33], which can only inpaint little part of the texture and structure information. Hence, given an input image $\mathcal{I}$, we first use the inpainting diffusion model $\mathcal{F}_{\theta_\mathcal{F}}$ to obtain an outpainted image $\hat{\mathcal{I}}$ that provides additional context for the scene beyond what was captured in the initial view:

$$\hat{\mathcal{I}} = \mathcal{F}_{\theta_\mathcal{F}}(\mathcal{I})\,. \tag{1}$$

This approach is particularly useful when the camera moves beyond the bounds of the original view. We also use the inpainting diffusion model to seamlessly inpaint the occluded regions.

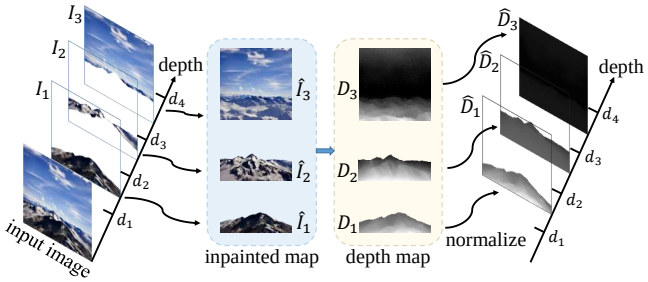

Figure 3: Layer-based Depth predict. We perform separate depth prediction, normalization, and remapping for each inpainted color layer, to ensure the correct depth relationship between layers.

3.3 Layer-based Representation

To fill in the occluded areas by diffusion model, we can intuitively refer to the previous method [1, 18, 19] that use rendering, refining, and repeating steps to inpaint missing content regions. However, since the output of the network is taken as a new input, small errors in each iteration may accumulate and eventually cause domain drifting, leading to poor output image quality. Therefore, we proposed *Layer-based Depth prediction* instead of repeat operation to represent the scene, which can effectively solve the above problems and guarantee the 3D consistency of the scene. We show the process in Fig. 3.

Given an outpainted image of the scene $\hat{\mathcal{I}}$, we divide the scene into different layers using depth information. To this end, we first use a pre-trained monocular depth estimation model [26] denoted as $\mathcal{D}_{\theta_\mathcal{D}}$ to predict the depth map $\hat{D}$ of the outpainted image $\hat{\mathcal{I}}$. Then, following [23], we use agglomerative clustering C to divide the depth range of the $\hat{D}$ into multiple intervals:

$$\{d_1, d_2, \cdots, d_i, \cdots, d_{L+1}\} = C(\hat{D})\,, \tag{2}$$

where L represents the number of layers after clustering. We separate $\hat{\mathcal{I}}$ into different layers based on depth intervals, where I_i denotes the i-th layered image with depth values between d_i and d_{i+1}. For an intermediate layer image I_i, it will produce many blank areas due to the occlusion of the foreground. To solve this problem, we use the diffusion model $\mathcal{F}_{\theta_\mathcal{F}}$ to fill these occluded areas in each layer, resulting in the inpainted layer maps $\hat{I}_i$:

$$\hat{I}_i = \mathcal{F}_{\theta_\mathcal{F}}(I_i)\,. \tag{3}$$

We compare the inpainting results obtained by different inpainting methods as shown in Fig. 4. One can see that the appearance of

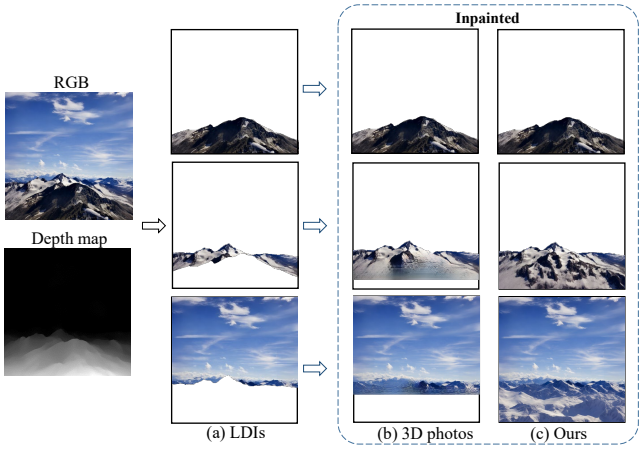

Figure 4: From an image to inpainted RGB layers. Given an input image and its estimated monocular depth [26], we first apply agglomerative clustering [23] to separate the RGB image into multiple (in this case, three) RGB layers using depth map as shown in (a), then (b) 3D Photos [33] perform context-aware inpainting to obtain inpainted layers. (c) shows the results of our inpainted layers using diffusion models.

the scene obtained by the diffusion model is more realistic, which helps us ensure the 3D consistency of the scene.

After obtaining the inpainted layer maps, we utilize the depth estimation model $\mathcal{D}_{\theta_\mathcal{D}}$ to generate the depth map of each layer. It is worth noting that the inpainted color layers are not a complete image (except for the last layer), as illustrated in Fig. 4(c), thus we cannot directly predict the corresponding depth map. Actually, we complement the incomplete inpainted layer with colors from the layers behind it. For any inpainted layer map $\hat{I}_i$, we predict its corresponding depth map D_i using the following formula:

$$D_i = \mathcal{D}_{\theta_\mathcal{D}}\left(\bigoplus_{s=i}^{L} \hat{I}_s\right), \tag{4}$$

where $\bigoplus$ denotes the cumulative overlay operation from the i-th layer to the L-th layer. Note that the shallower layers with lower depth values will overlay the deeper layers with higher depth values. Since each layer uses a separate depth estimation, the original depth relationship between layers will be misaligned, resulting in incorrect 3D representations. Therefore, we extract and normalize the depth values from all depth layers, and remap them to the value interval of the corresponding depth layer D_i. This process can be computed by:

$$\hat{D}_i = \frac{(d_{i+1} - d_i) \cdot (\hat{D}_i - min(\hat{D}_i))}{max(\hat{D}_i) - min(\hat{D}_i)} + d_i\,, \tag{5}$$

Through the *Layer-based Depth prediction*, our method can maintain the 3D consistency of the scene without using repeated operations.

3.4 Scene Animation and Image Rendering

To create a 3D-consistent representation of the scene, we first introduce a 2D feature extraction network [38] to encode features of inpainted color layers. Then we unproject the features into 3D using their corresponding depth layers, resulting in a feature point cloud $\mathcal{P} = \{(X_i, f_i)\}$, where X_i and f_i are 3D coordinates and the feature vector for each 3D point respectively. To animate dynamic objects, we estimate a motion field for the observed scene. Following Holynski et al. [12], we assume that a time-invariant and constant-velocity motion field, termed Eulerian flow field, can well approximate the bulk of real-world motions, such as clouds, smoke, and water. Formally, we denote $F_{t\to t+1}(\cdot)$ as the Eulerian flow field of the scene, which represents how each pixel in the t frame moves to the $t+1$ frame. Specifically, we begin by estimating the 2D motion field $F_{t\to t+1}(\cdot)$ from the outpainted image using a pre-trained image-to-image translation network [12], and elevate this motion field into 3D scene flow at time t with the aid of estimated depth values. Then, we move each 3D point by calculating its destination as its original position plus the scene flow. However, as points move forward, increasingly large holes can appear in the displaced point cloud. This happens when points move out of their original positions without any other points filling in those unknown regions. To address this issue, we employ the 3D symmetric animation [16] to fill in the holes as points move forward. This allows us to obtain the final displaced point cloud $\mathcal{P}_m(t) = \{(X_i^m(t), f_i)\}$ at time t.

We also obtain the camera pose by adopting the autocruise algorithm from [19]. The autocruise algorithm is particularly effective as it can adaptively adjust the camera trajectory according to the scene information. Therefore, we can use this algorithm to obtain a reasonable end-frame camera pose c_N from the view of the input image c_1, and generate a continuous camera motion trajectory by interpolating the two camera poses to N intermediate values.

After obtaining the animated feature point clouds and camera poses, our final step is to render them into output images. As we outpaint the initial input image to get a higher resolution and larger field of view, we adjust the camera intrinsics at each frame to ensure that the resolution of the output image is consistent with the input image. We then use a differentiable point-based renderer [39] to splat the displaced point clouds into the target image plane, maintaining the original resolution of the input image. As a result, we can render the displaced point clouds $\mathcal{P}_m(t)$ at time t into an image. By rendering displaced point clouds at all times, we obtain a series of rendered frames that can be compiled into the final coherent video.

4 EXPERIMENTS

4.1 Baselines

To evaluate the effectiveness of our method, it is essential to compare it against the current state-of-the-art models. However, since our approach involves animation alongside long-range view generation, there are no previous works that have explored this specific combination. Therefore, a direct comparison with previous works is not possible. To overcome this challenge, we compare our method with the state-of-the-art models in the respective areas of long-range view generation and 3D animation. For long-range view generation, we use InfNat-zero [19], which has achieved remarkable results in generating high-quality and diverse long-range views. For 3D animation, we use 3D Cinemagraphy [16], which can produce plausible animation of the scene while allowing slight

Figure 5: Qualitative comparisons of baselines and our method on LHQ dataset. From left to right, we show generated views over trajectories of length 100 for three methods: InfNat-zero [19], 3D Cinemagraphy [16], and ours.

camera movements. As for T2V methods, they cannot achieve the goal of our work, which is to control the camera pose to generate new dynamic viewpoint images. Therefore, in our experimental settings, we cannot compare the metrics such as PSNR, SSIM, and LPIPS with them due to the inability of T2V methods to generate frames corresponding to the camera poses for comparison with the ground truth frames.

4.2 Results

Evaluation dataset. Following [19], we evaluate our method and baselines using two public datasets of nature scenes: the Landscape High Quality (LHQ) dataset [37], a collection of 90K nature landscape photos, and the Aerial Coastline Imagery Dataset (ACID) [19], a video dataset of nature scenes with SfM camera poses. Since there is no multi-view data for LHQ dataset, we only do qualitative experiments on it. The ACID dataset contains 279 evaluation sequences, where each sequence has an input frame and subsequent ground truth frames and the corresponding camera pose for each frame.
Evaluation metrics. We adopt PSNR (Peak Signal to Noise Ratio), SSIM (Structural Similarity Index Measure), and Perceptual Similarity (LPIPS) [40] as our evaluation metrics. Besides, we measure multi-view consistency using photometric error, which measures the $\mathcal{L}_1$ error when backward warping the result after one camera step to the initial frame and multiply it by 100. To ensure that dynamic objects' movement does not interfere with the consistency measurements, we keep the dynamic objects stationary during photometric error calculation. All methods are evaluated at 512×512.
Quantitative comparisons. We show the quantitative metrics of our method against baselines on ACID evaluation sequences in Table 1. Our method outperforms the other baselines in terms of view generation on all metrics, achieving highly competitive performance. Specifically, our approach achieves the highest PSNR and SSIM scores, which indicate that the generated views have high fidelity and are perceptually similar to the ground truth views. Moreover, our method also has the lowest LPIPS score, which indicates that our generated views are more visually similar to the ground truth views compared to the other baselines. These quantitative results demonstrate the effectiveness of our approach in generating high-quality views compared to the baseline models.
Qualitative comparisons. Visual qualitative comparisons are shown in Fig. 5. InfNat-zero quickly degenerates and leads to poor 3D consistency and semantic consistency of foreground contents due to its per-frame generation protocols and lack of underlying 3D scene representation, giving a very strong sense of unreality. Also,

Table 1: Quantitative comparisons on ACID evaluation sequences. Our method outperforms all baselines in all metrics, indicating that our method achieves better perceptual quality and produces a more realistic rendering.

Method	PSNR↑	SSIM↑	LPIPS↓	consistency↓
InfNat-zero [18]	20.23	0.568	0.364	5.36
3D Cinemagraphy [16]	21.29	0.596	0.316	1.92
Ours	**22.72**	**0.634**	**0.273**	**1.12**

it cannot animate dynamic objects in the scene such as clouds. 3D Cinemagraphy suffers from severe artifacts and voids in long-term view generation. Specifically, its rough depth inpainting causes adhesion between the different layers and leads to artifacts and holes. In the second and fifth rows, the simple texture color inpainting of 3D Cinemagraphy results in a lack of semantic information in its fills, and because it can only partially fill the image, it results in voids in the image. Moreover, the content is missing at the image boundaries after moving the camera because there is no external expansion of the image. Our method, in contrast, maintains high 3D consistency and demonstrates significantly improved synthesis quality and realism. We encourage readers to watch our supplementary video for a better visual comparison.

Text driven generation. By using a pre-trained diffusion model that is conditioned on the user's text prompt, we can effectively use the prompt to guide the image inpainting process. Thus, we can generate diverse scenes that the users expect. Fig. 6 illustrates the differentiated output based on text. We also provide a supplementary video that demonstrates our method in more detail, showing how the text prompts can be used to generate different types of scenes and how our method produces realistic and visually appealing results.

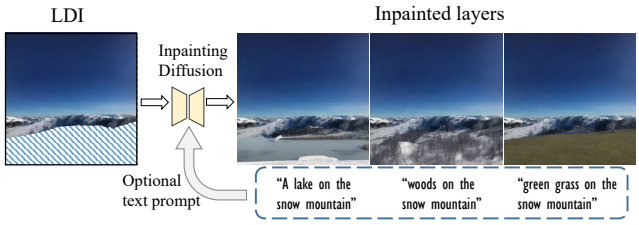

Figure 6: Text-driven inpainting. By using different text prompts as guidance, it is possible to generate different content that can provide the desired scene experience to the user.

Flying out of the input image. Our main objective is to generate a long sequence video that simulates the visual effect of flying into an input image. Additionally, our proposed method can also generate flying-out videos by moving the camera position backward, as demonstrated in Fig. 7. This additional capability of our method enhances its versatility and usefulness in various applications. Readers are encouraged to view our supplementary video for better visual experiences.

Generalization on in-the-wild photos. Unlike previous works that are only applicable to restricted domains, such as landscapes,

Figure 7: Flying out of the input image. By moving the camera backward, our method can generate the fly-out effect.

due to the fact that they are trained on scenario-specific datasets, our framework can generate long-range novel views in diverse, in-the-wild scenes without training. Furthermore, existing approaches for long-term view synthesis, such as [1, 18], can only render images at 128×128 resolution and require super-resolution networks to achieve higher resolution. In contrast, our method can be directly applied to any resolution. As illustrated in Fig. 8, we present flythrough results on paintings and historical photos at 640×1024 resolution, which demonstrates the robust generalization capabilities of our approach.

4.3 Ablation Study

Each component of our system plays an important role in improving the rendering quality. To justify our design choices, we conduct ablation studies, as presented in Table 2. Visual results of the ablation study are shown in Fig. 9. In the "w/o outpainting" experiment, we demonstrate that outpainting significantly improves the performance by providing realistic and reliable supplementary content when the camera moves out of the initial view boundaries. Without outpainting, there would be many holes around the image borders, as depicted in the second column of the figure. In the "w/o inpainting" experiment, we show that if the layered depth image is not inpainted, the rendered image will exhibit holes at the depth discontinuity when the camera moves. Finally, in the "per-frame generation" experiment, we adopt the same strategy as InfNat-zero, where we move the camera position slightly and generate one image at a time. We then use this image as the new input and repeat the process to obtain a coherent video. As we can see, adopting such a strategy leads to poor consistency and rendering quality.

Table 2: Ablation Study. Each component of our system leads to an increase in the rendering quality.

	PSNR↑	SSIM↑	LPIPS↓	Consistency↓
w/o outpainting	22.29	0.612	0.289	1.63
w/o inpainting	21.23	0.591	0.335	1.96
per-frame generation	20.51	0.577	0.351	3.35
Full model	**22.72**	**0.634**	**0.273**	**1.12**

Figure 8: Generalization of our method. Our method can work at any resolution and can be generalized to paintings and historical photos in addition to real-world photos.

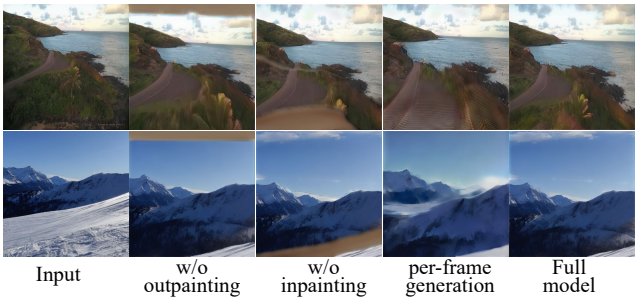

Figure 9: Visual examples of the ablation study. Each row shows the results in different settings, presented from left to right in the following sequence: input view, results without outpainting, results without inpainting, results using per-frame generation and results obtained from our full model.

4.4 User Study

To better evaluate the perceptual quality and realism of our method in the view of humans, we conduct a user study to compare it with all baselines. Specifically, we collect 30 photos from the LHQ dataset and ACID evaluation sequences. We use different approaches to generate videos with identical settings. During the study, participants were shown an input image and two animated videos, one generated by our method and another randomly selected approach, in random order. 107 volunteers were invited to choose the method with better perceptual quality and realism or choose none if they found it difficult to judge. Our results, presented in Table 3, indicate that our method outperforms alternative approaches by a significant margin in terms of realism and immersion.

Table 3: User study. The results indicate that our method is preferred by users as it offers a more realistic and immersive experience compared to alternative approaches.

Comparison	Human preference
InfNat-zero [18] / **Ours**	3.4%/**96.6%**
3D Cinemagraphy [16] / **Ours**	17.2%/**82.8%**

5 CONCLUSION

We present a novel approach for synthesizing long-term flythrough videos of dynamic scenes from a single image, maintaining 3D consistency, and producing realistic output videos without the need for large-scale training. Our framework is flexible, allowing for both flying in and flying out of the input image and customization based on user-provided text prompts. Extensive experiments demonstrate the effectiveness of our method. A user study is also conducted to validate the compelling rendering results of our method. We hope that our work can offer a new direction for consistent long-term dynamic scene video synthesis from a single image and inspire further research in the field.

Limitations and future work. Our approach focuses on filling in hierarchical occlusion information, but may not complement vertical information, such as obtaining a more detailed image when the camera moves forward. Additionally, if the depth prediction module estimates the wrong geometry from the input image, such as wrong layering, our method may not work well. Moreover, our method primarily focuses on handling common moving elements like fluids, while more complex object movements remain a research problem. We plan to explore more effective solutions to address these limitations in future work.

ACKNOWLEDGMENTS

This work was funded by the National Natural Science Foundation of China under Grant No.U1913602 and was partly supported by the RIE2020 Industry Alignment Fund - Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

REFERENCES

[1] Shengqu Cai, Eric Ryan Chan, Songyou Peng, Mohamad Shahbazi, Anton Obukhov, Luc Van Gool, and Gordon Wetzstein. 2022. DiffDreamer: Consistent Single-view Perpetual View Generation with Conditional Diffusion Models. *arXiv preprint arXiv:2211.12131* (2022).

[2] Lucy Chai, Richard Tucker, Zhengqi Li, Phillip Isola, and Noah Snavely. 2023. Persistent Nature: A Generative Model of Unbounded 3D Worlds. *arXiv preprint arXiv:2303.13515* (2023).

[3] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. 2019. Everybody dance now. In *Proceedings of the IEEE/CVF international conference on computer vision*. 5933–5942.

[4] Zhaoxi Chen, Guangcong Wang, and Ziwei Liu. 2023. Scenedreamer: Unbounded 3d scene generation from 2d image collections. *arXiv preprint arXiv:2302.01330* (2023).

[5] Yung-Yu Chuang, Dan B Goldman, Ke Colin Zheng, Brian Curless, David H Salesin, and Richard Szeliski. 2005. Animating pictures with stochastic motion textures. In *ACM SIGGRAPH 2005 Papers*. 853–860.

[6] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. 2015. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*. 2758–2766.

[7] Yuki Endo, Yoshihiro Kanamori, and Shigeru Kuriyama. 2019. Animating landscape: self-supervised learning of decoupled motion and appearance for single-image video synthesis. *arXiv preprint arXiv:1910.07192* (2019).

[8] Siming Fan, Jingtan Piao, Chen Qian, Kwan-Yee Lin, and Hongsheng Li. 2022. Simulating Fluids in Real-World Still Images. *arXiv preprint arXiv:2204.11335* (2022).

[9] Rafail Fridman, Amit Abecasis, Yoni Kasten, and Tali Dekel. 2023. Scenescape: Text-driven consistent scene generation. *arXiv preprint arXiv:2302.01133* (2023).

[10] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. 2022. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303* (2022).

[11] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. 2022. Video diffusion models. *arXiv preprint arXiv:2204.03458* (2022).

[12] Aleksander Holynski, Brian L Curless, Steven M Seitz, and Richard Szeliski. 2021. Animating pictures with eulerian motion fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5810–5819.

[13] Wei-Cih Jhou and Wen-Huang Cheng. 2015. Animating still landscape photographs through cloud motion creation. *IEEE Transactions on Multimedia* 18, 1 (2015), 4–13.

[14] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. 2023. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. *arXiv preprint arXiv:2303.13439* (2023).

[15] Jing Yu Koh, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. 2021. Pathdreamer: A world model for indoor navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14738–14748.

[16] Xingyi Li, Zhiguo Cao, Huiqiang Sun, Jianming Zhang, Ke Xian, and Guosheng Lin. 2023. 3D Cinemagraphy from a Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4595–4605.

[17] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. 2018. Flow-grounded spatial-temporal video prediction from still images. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 600–615.

[18] Zhengqi Li, Qianqian Wang, Noah Snavely, and Angjoo Kanazawa. 2022. Infinitenature-zero: Learning perpetual view generation of natural scenes from single images. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*. Springer, 515–534.

[19] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. 2021. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14458–14467.

[20] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. 2019. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5904–5913.

[21] Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. 2023. VideoFusion: Decomposed Diffusion Models for High-Quality Video Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10209–10218.

[22] Aniruddha Mahapatra and Kuldeep Kulkarni. 2022. Controllable Animation of Fluid Elements in Still Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3667–3676.

[23] Oded Maimon and Lior Rokach. 2005. Data mining and knowledge discovery handbook. (2005).

[24] Juewen Peng, Jianming Zhang, Xianrui Luo, Hao Lu, Ke Xian, and Zhiguo Cao. 2022. Mpib: An mpi-based bokeh rendering framework for realistic partial occlusion effects. In *European Conference on Computer Vision*. Springer, 590–607.

[25] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022).

[26] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. 2021. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 12179–12188.

[27] Xuanchi Ren and Xiaolong Wang. 2022. Look outside the room: Synthesizing a consistent long-term 3d scene video from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3563–3573.

[28] Yurui Ren, Xiaoming Yu, Junming Chen, Thomas H Li, and Ge Li. 2020. Deep image spatial transformation for person image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7690–7699.

[29] Chris Rockwell, David F Fouhey, and Justin Johnson. 2021. Pixelsynth: Generating a 3d-consistent experience from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14104–14113.

[30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10684–10695.

[31] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems* 35 (2022), 36479–36494.

[32] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. 1998. Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. 231–242.

[33] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 2020. 3d photography using context-aware layered depth inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8028–8038.

[34] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. 2019. First order motion model for image animation. *Advances in Neural Information Processing Systems* 32 (2019).

[35] Aliaksandr Siarohin, Oliver J Woodford, Jian Ren, Menglei Chai, and Sergey Tulyakov. 2021. Motion representations for articulated animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13653–13662.

[36] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. 2022. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792* (2022).

[37] Ivan Skorokhodov, Grigorii Sotnikov, and Mohamed Elhoseiny. 2021. Aligning latent and image spaces to connect the unconnectable. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14144–14153.

[38] Qianqian Wang, Zhengqi Li, David Salesin, Noah Snavely, Brian Curless, and Janne Kontkanen. 2022. 3D moments from near-duplicate photos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3906–3915.

[39] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. 2020. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7467–7477.

[40] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 586–595.