SVG: 3D Stereoscopic Video Generation via Denoising Frame Matrix

Peng Dai[1,2]   **Feitong Tan**[1]* **Qiangeng Xu**[1]*   **David Futschik**[1]
Ruofei Du[1]   **Sean Fanello**[1]   **Xiaojuan Qi**[2]   **Yinda Zhang**[1]
[1]Google   [2]The University of Hong Kong

Abstract

Video generation models have demonstrated great capabilities of producing impressive monocular videos, however, the generation of 3D stereoscopic video remains under-explored. We propose a pose-free and training-free approach for generating 3D stereoscopic videos using an off-the-shelf monocular video generation model. Our method warps a generated monocular video into camera views on stereoscopic baseline using estimated video depth, and employs a novel *frame matrix* video inpainting framework. The framework leverages the video generation model to inpaint frames observed from different timestamps and views. This effective approach generates consistent and semantically coherent stereoscopic videos without scene optimization or model fine-tuning. Moreover, we develop a disocclusion boundary re-injection scheme that further improves the quality of video inpainting by alleviating the negative effects propagated from disoccluded areas in the latent space. We validate the efficacy of our proposed method by conducting experiments on videos from various generative models, including Sora [4], Lumiere [2], WALT [8], and Zeroscope [42]. The experiments demonstrate that our method has a significant improvement over previous methods. The code will be released at https://daipengwa.github.io/SVG_ProjectPage/

1   Introduction

As VR/AR technology advances, the demand for creating stereoscopic content and delivering immersive 3D experiences to users continues to grow. Due to visual sensitivity, binocular stereoscopic content should feature flawless 3D and semantic consistency between both eye views, as well as seamless temporal consistency across frames. While monocular video generation models have been extensively researched and methods are now capable of synthesizing high-fidelity videos that adhere to complex text prompts [4], there has not been much progress in the realm of generating 3D stereoscopic videos at the scene level. One reason for this gap lies in the substantial amount of monocular video data that is readily available, contrasted with the scarcity of stereo video data for training models to generate stereoscopic videos directly.

An emergent solution is to convert generated monocular videos into stereoscopic videos using novel view synthesis [24; 27]. However, these methods usually overly rely on camera pose estimation, which is a challenging task on its own either using SFM [39] or joint optimization [27], and as a result tend to be unstable, particularly in dynamic scenes where cameras experience subtle motions or when the content is dominated by dynamic objects with temporally varying appearances, both of which are prevalent in generated videos. Consequently, these methods fail in optimizing 3D scenes and offer low-quality solutions to the task (see Fig. 3). Moreover, these approaches are based on reconstruction, lacking the generative ability to hallucinate occluded regions in the novel views that do not appear in any of the remaining video frames.

---

*Equal contribution

In this paper, we propose an alternative pose-free and training-free framework, for the sake of robustness and generalization capability, that operates solely by exploiting inference of an off-the-shelf video generation model [42] to generate high quality 3D stereoscopic videos. Our initial attempt follows a typical 2D to 3D image uplifting methodology [14] and extends it into the video domain. Specifically, we first generate a monocular video as the left view, which is then reprojected into the right view using per-frame estimated monocular depths [46], where we apply temporal-spatial smoothing to improve the consistency of the estimated depth. Subsequently, we leverage an off-the-shelf video generation model's [42] ability to generate natural videos, by adding noise and denoising the warped video frames to inpaint the disoccluded regions, inspired by diffusion-based image inpainting [1].

However, this naive pipeline does not produce appealing results: inpainting the right-view video frames independently, without referencing the left view, typically generates semantically mismatched content. To address this problem, we propose a novel representation, called the *frame matrix*, which contains frame sequences observed from a number of viewpoints evenly distributed along the baseline between two eyes. The frame sequences along the view direction (rows of the matrix) form videos with camera motion, while the frame sequences along the time direction (columns of the matrix) form videos with scene motions (see Fig. 1 second column). Since the video diffusion model has video prior for both scene and camera motions, we propose to jointly update the entire *frame matrix* from both directions. In each denoising step, we use resample techniques [28] by alternatively denoising frame sequences along the view and the time directions. Finally, we obtain a semantically consistent and temporally smooth 3D stereoscopic video by taking the leftmost and the rightmost frame sequences to represent the left-eye view and the right-eye view, respectively.

Furthermore, we note that the inevitable resolution downsampling operation in most video generation models with latent encoding [4; 2; 42; 8] is detrimental to the video inpainting task. During encoding, the dark pixels created by disocclusion can degrade the features near the disocclusion boundary, leading to undesirable artifacts (see Fig. 5). Instead of following the inpainting scheme proposed in previous work [1], which encodes the latent feature only once, we iteratively update both the disoccluded regions in the image space and the latent feature map with generated content during the diffusion process. This approach re-injects the generated content into the disocclusion boundary, which mitigates the negative impact of dark disocclusion and effectively prevents the artifacts.

To validate the efficacy of our proposed method, we generate stereoscopic video from monocular videos generated by Sora [4], Lumiere [2], WALT [8], and Zeroscope [42]. Both qualitative and quantitative evaluations suggest that our approach outperforms other baselines in 3D stereoscopic video generation. Our contributions are summarized as follows:

- We design a novel pipeline to generate 3D stereoscopic videos. Unlike previous work, our method does not need camera pose estimation or fine-tuning on specific datasets.
- We propose a novel *frame matrix* representation that regularizes the diffusion-based video inpainting to generate semantically consistent and temporally smooth content.
- We propose a re-injection scheme that drastically reduces the negative influence of disoccluded regions in latent space and produces high-quality results.
- We conduct comprehensive experiments that show the superiority of our approach over previous methods for 3D stereoscopic video generation.

2 Related Work

Video Generation. Video generation [42; 4; 2; 8; 9; 11; 13; 40] has achieved tremendous progress since the advent of the diffusion model [12]. Taking into account the dataset requirements and scarcity of tagged videos, a prominent approach for video generation is to extend pre-trained image generation models [37; 38; 36] by inserting additional temporal layers and then fine-tuning them on video data [7; 3; 44]. To further improve the compute efficiency and enable long clip processing, WALT [8] and Lumiere [2] proposed to compress the video in both the temporal and spatial dimensions. More recently, Sora [4] adopted a transformer diffusion architecture [34] and was trained on large-scale video datasets to produce impressive video generation results. Different from previous video generation models focusing on producing higher-quality and longer monocular videos, our method orthogonally explores the possibility of leveraging pre-trained video generation models for stereoscopic 3D video generation.

Novel View Synthesis. Great progress has been made for novel view synthesis in both static and dynamic scenes captured by single or multiple cameras [30; 47; 20; 17; 31]. Mildenhall *et al.* [30] proposed to encode the static scene into neural radiance fields (NeRF), which were then used for novel view synthesis through volume rendering. For more challenging scenes with dynamic content, follow-up works additionally optimized a deformation field [32; 15; 33] or scene flow fields [23] to handle the motion of dynamic objects. Instead of encoding the scene into a NeRF, DynIBaR [24] leveraged nearby frames for rendering novel view images, and dynamic objects were handled by optimized motion fields. Different from methods requiring pre-computed camera poses, RoDynRF [27] jointly optimized the NeRF and camera poses from scratch. Concurrently, FVS [19] achieves novel view video synthesis using a plane-based scene representation. Although these approaches produce high-quality renderings, they are limited to scenes where the camera pose can be accurately estimated and have limited synthesis capability. In contrast, we design a method that explicitly avoids having to estimate camera poses and possesses the ability to hallucinate unseen content.

3D Content Creation and Inpainting. Automated 3D content creation [14; 5; 6; 48] is another related area, with emerging approaches such as inpainting [11] or multi-view generators [26; 43]. Recently, Text2Room [14] proposed creating a 3D room by warping an image into novel views and using a text-guided inpainter to deal with disocclusions. WonderJourney [48] made this process automatic by including a large language model in the loop. Similar to creating static scenes, we could use pretrained video inpainter [50; 22] for dynamic 3D content creation, however, these models suffer from generalization problems in creating high-quality, consistent 3D content. Lastly, Deep3D [45] is trained using 3D movies, with the goal of converting 2D videos into stereoscopic videos. However, the training data is not publicly available and it lacks the flexibility to modify videos for creative purposes, such as different stereo baselines. In this paper, we explore the possibilities of using video generation models for 3D video creation without training on specific, hard-to-obtain datasets.

3 Stereoscopic Video Generation

Conditioned on a text prompt or a single image c, our method aims to generate 3D stereoscopic video $\{\mathbf{X}_l, \mathbf{X}_r\}$, consisting of two monocular sequences. The most straightforward way is to use a diffusion-based generation model $\mathcal{G}$:

$$\{\mathbf{X}_l, \mathbf{X}_r\} = \mathcal{G}(\{\epsilon_t | t = 1, ..., T\}, c), \tag{1}$$

where $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the sampled noise at step t. The generated stereoscopic videos should possess the following characteristics: First, the appearance and semantics between the left eye view $\mathbf{X}_l$ and right eye view $\mathbf{X}_r$ should be consistent and be temporally stable. Second, the stereo effect should be prominent and immersive. Last, the generated content should be diverse and controllable with the given conditioning.

However, training a $\mathcal{G}$ that can directly generate stereo videos $\{\mathbf{X}_l, \mathbf{X}_r\}$ with the desired properties requires a vast dataset of stereo videos with diverse content. Due to the scarcity of such data, we propose a training-free approach that relies on an off-the-shelf depth estimator [46] and a diffusion-based monocular video generation model $\mathcal{G}$ such as Zeroscope [42]. We first generate a monocular video for one eye using a video diffusion model [42; 8; 4; 2] (Eq. 2), then obtain the other video view by conditioning on the first video. To automatically preserve 3D consistency, we implement this conditioning by estimating depth $\mathbf{d}_l$ for the left video and warp its content to obtain the right view sequence $\mathbf{X}_{l \to r}$ with disocclusion masks $\mathbf{M}_r$ (Eq. 3) according to the stereoscopic baseline. Then, we use $\mathcal{G}$ again to inpaint the disoccluded parts by denoising inpainting process [1; 28] (Eq. 4), obtaining the other eye view video $\mathbf{X}_r$.

$$\mathbf{X}_l = \mathcal{G}(\{\epsilon_t | t = 1, ..., T\}, c), \tag{2}$$

$$\mathbf{X}_{l \to r}, \mathbf{M}_r = \text{Warp}_{l \to r}(\mathbf{X}_l, \mathbf{d}_l), \tag{3}$$

$$\mathbf{X}_r = \mathcal{G}(\{\epsilon_t | t = 1, ..., T\}, c, \mathbf{X}_{l \to r}, \mathbf{M}_r). \tag{4}$$

In Sec. 3.1, we describe the video depth warping. In Sec. 3.2, we introduce the *frame matrix* representation for the video inpainting. Our denoising frame matrix drastically improves the semantic similarity between $\mathbf{X}_l$ and $\mathbf{X}_r$ and helps preserve temporal smoothness. Last but not least, a disocclusion boundary re-injection mechanism is introduced to further improve the inpainting quality in Sec. 3.3. An overview of our method is displayed in Fig. 1.
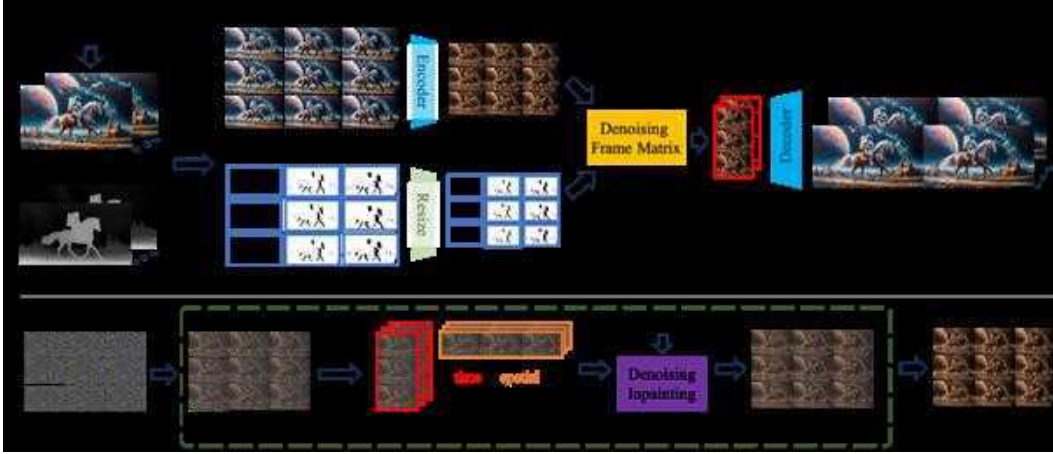
3

Figure 1: **Overview – Top:** Given a text prompt, our method first uses a video generation model to generate a monocular video, which is warped (using estimated depth) into pre-defined camera views to form a *frame matrix* with disocclusion masks M. Then, the disoccluded regions are inpainted by denoising the frame sequences within the *frame matrix*. After denoising, we select the leftmost and the rightmost columns and decode them to obtain a 3D stereoscopic video. **Bottom:** Details of denoising *frame matrix*. We initialize the latent matrix $\mathbf{z}_T$ as a random noise map. For each noise level, we extend the resampling mechanism [16; 28] to alternatively denoise temporal (column) sequences and spatial (row) sequences N times. Each time, row or column sequences are denoised and inpainted (see Fig.2). By denoising along both spatial and temporal directions, we obtain an inpainted latent $\mathbf{z}_0$ which can be decoded into temporally smooth and semantically consistent sequences.

3.1 Monocular Video Depth Warping

The depth estimation model [46] is applied to predict all frames' depth values, which will be smoothed to produce more consistent video depths. Specifically, we utilize the estimated optic flows [41] to align consecutive depth frames. The outliers in predicted depths will be suppressed by convolving with a Gaussian kernel along the time axis. After obtaining RGB-D frames, we can warp them into target camera views where disoccluded regions appear. In addition, the warped images usually contain isolated pixels, and the foreground and background are entangled, which jeopardizes video quality [5]. To handle these problems, we follow Dai *et al.* [5] to project points into multi-plane images [51], then remove isolated pixels and cracks and finally obtain a noisy-points-free image. (See supplemental material for details).

3.2 Video Inpainting with Frame Matrix

The inpainting pipeline plays a key role in ensuring spatial/semantic and temporal consistency. While image inpainting approaches [1; 28] provide a reasonable baseline, the results lack temporal and spatial stability. Therefore, we introduce a Frame Matrix representation, which addresses both issues.

Single Video Denoising Inpainting   Inspired by RePaint [28], we extend the diffusion-based image inpainting to video inpainting. We use the video generation model $\mathcal{G}$ (i.e., Zeroscope [42]) as our inpainting tool, which is a latent diffusion model consisting of a VAE encoder $\mathcal{E}$, a decoder $\mathcal{D}$ and a latent denoiser $\{\epsilon_\theta, \Sigma_\theta\}$. First, the warped video is fed into the VAE encoder to obtain video latent features $\mathbf{z}_0^{\text{known}} = \mathcal{E}(\mathbf{X}_{l\rightarrow r})$. Then, we resize the image disocclusion masks $\mathbf{M}_r$ to the resolution of the latent and obtain latent disocclusion masks $\mathbf{m}$. During the denoising process, we start from a random noisy latent map $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. For each subsequent step t, we sample a new intermediate noisy latent map from $\mathbf{z}_0$ (Eq. 5), denoises the latent map from the last step $\mathbf{z}_t$ (Eq. 6) and combine them with $\mathbf{m}$ to obtain the $\mathbf{z}_{t-1}$ (Eq. 7). We visualize the following steps in Fig.2 (b):

$$\mathbf{z}_{t-1}^{\text{known}} \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\mathbf{z}_0^{\text{known}}, (1-\bar{\alpha}_t)\mathbf{I}\right), \tag{5}$$

$$\mathbf{z}_{t-1}^{\text{unknown}} \sim \mathcal{N}\left(\frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{z}_t, c, t)\right), \Sigma_\theta(\mathbf{z}_t, c, t)\right), \tag{6}$$

$$\mathbf{z}_{t-1} = m \odot \mathbf{z}_{t-1}^{\text{known}} + (1-m) \odot \mathbf{z}_{t-1}^{\text{unknown}}, \tag{7}$$
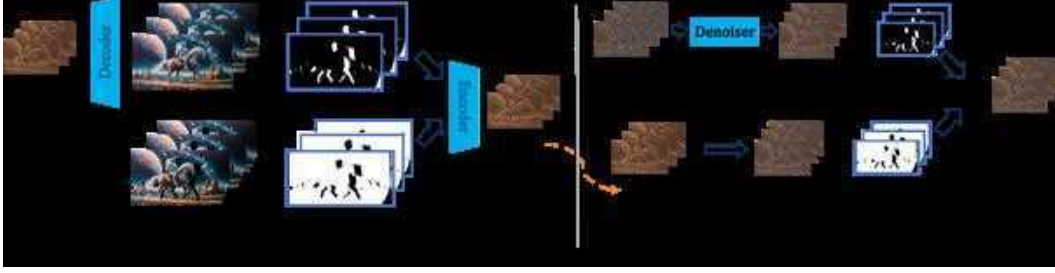
4

Figure 2: **Denosing Inpainting**. This figure visualizes the operations in the purple box of Fig.1. (a) We re-inject the generated content from a denoised latent $\tilde{\mathbf{z}}_0$ to update $\mathbf{z}_0^{known}$ and reduce its feature corruption on the disocclusion boundary. (b) A noisy latent $\mathbf{z}_t$ is denoised to $\mathbf{z}_{t-1}^{unknown}$. We take its disoccluded region and combine it with the unoccluded region of $\mathbf{z}_0^{known}$.

where $\bar{\alpha}_t$ and β_t denote the total noise variance and one step noise variance at t, respectively; $\epsilon_\theta(\mathbf{z}_t, c, t)$ and $\Sigma_\theta(\mathbf{z}_t, c, t)$ are predicted noise and variance for noisy latent map at $t-1$ step. Finally, we can obtain the inpainted right view sequence X_r by decoding the denoised latent $X_r = \mathcal{D}(\mathbf{z}_0)$.

By applying the above video inpainting scheme for the right view, we implement Eq. 4 and successfully hallucinate the disoccluded (unknown) regions while preserving the unoccluded (known) regions. The video diffusion model also ensures temporal smoothness. However, the inpainted content on the right view usually lacks semantic consistency w.r.t. the left view, as shown in the third column of Fig. 4. This is because we only condition on the left view by depth warping, while dropping the conditioning during inpainting.

Frame Matrix Representation. We propose a novel representation–*frame matrix*, which targets consistent dynamic content generation across space and time. As shown in Fig. 1 top, it is a matrix consisting of multiple frames, where each row represents frames observed from different camera poses at the same time stamp, and each column is a video recorded by a fixed camera at different time stamps. Consequently, the frame matrix can be defined as:

$$\mathbf{X} \equiv \left[\begin{array}{ccc} | & & | \\ \mathbf{x}_{(:,0)} & \cdots & \mathbf{x}_{(:,V)} \\ | & & | \end{array} \right] \equiv \left[\begin{array}{ccc} - & \mathbf{X}_{(0,:)} & - \\ & \vdots & \\ - & \mathbf{X}_{(S,:)} & - \end{array} \right]$$

where S and V are the largest indices of time steps and views, respectively. A view sequence (row) $\mathbf{X}_{(s,:)}$ forms a video with camera motions, while a time sequence (column) $\mathbf{X}_{(:,v)}$ forms a video with time-varying scene motions. Since the video diffusion model can denoise a sequence to a temporally and semantically consistent video, jointly denoise the rows and columns can ensure consistency spatially and temporally. Finally, we can obtain a 3D stereoscopic video by taking the leftmost and the rightmost time sequences $\mathbf{X}_{(:,0)}, \mathbf{X}_{(:,V)}$.

Constructing Frame Matrix. We evenly add V camera views distributed along the baseline between the two eyes with the same orientation of the reference view. Then, we warp the refence video (the $0th$ column) based on depth (Sec. 3.1) into these views and obtain $\mathbf{X}_{warp} \equiv [\mathbf{X}_{(:,0)}, \mathbf{X}_{(:,0 \to 1)}, ..., \mathbf{X}_{(:,0 \to V)}]$ with a disocclusion masks matrix $\mathbf{M}$.

Denoising Frame Matrix. Similar to single video sequence inpainting, we encode frame matrix into a latent frame matrix $\mathbf{z}_0^{known} = \mathcal{E}(\mathbf{X}_{warp})$, and resize $\mathbf{M}$ to obtain latent disocclusion map $\mathbf{m}$. We also initialize $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. As shown in Fig.1 (Bottom), for each noise level, we extend the resampling mechanism [28] to alternatively denoise column sequences and row sequences N times. Each time, row or column sequences are denoised following Eq. 5-7 and we add back noise between every resampling iteration:

$$\mathbf{z}_t \sim \mathcal{N}(\sqrt{1 - \beta_{t-1}}\mathbf{z}_{t-1}, \beta_{t-1}\mathbf{I}). \tag{8}$$

Please refer to Sec.A in the supplemental material. By denoising along these two directions alternatively, the spatial and temporal sequences will proceed toward a harmonic state in the end.

3.3 Disocclusion Boundary Re-Injection

Since most video generation models are using latent diffusion, the disoccluded dark regions of $\mathbf{X}_{warp}$ will be propagated beyond the latent mask $\mathbf{m}$ during VAE encoding (*e.g.*, Zeroscope downsamples

by $8\times$), leading to defective latent features on $\mathbf{z}_0^{\text{known}}$'s disocclusion boundary. This would lead to artifacts in the final results (Fig. 5 left).

We propose to re-inject the denoised information in the disoccluded regions to improve the latents on this boundary. Specifically, we predict the denoised latent features [12], which are decoded into a denoised video (Eq. 9). Then, we replace its unoccluded regions with warped pixels to form a video that is faithful to the reference view but with better disocclusion pixels. By encoding this video, we can get a updated $\mathbf{z}_0^{\text{known}}$ (Eq. 10) which alleviates corruption on the boundary:

$$\widetilde{\mathbf{X}}_0 = \mathcal{D}(\widetilde{\mathbf{z}}_0), \text{where } \widetilde{\mathbf{z}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{z}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(\mathbf{z}_t, c, t) \right), \tag{9}$$

$$\mathbf{z}_0^{\text{known}} = \mathcal{E} \left(\mathbf{M} \odot \mathbf{X}_{warp} + (1 - \mathbf{M}) \odot \widetilde{\mathbf{X}}_0 \right). \tag{10}$$

After this, this improved $\mathbf{z}_0^{\text{known}}$ can be used in Eq. 5 for the next iteration.

4   Experiments

Datasets.   To validate the effectiveness of our method, we conduct experiments using a variety of recent video generation models, including Sora [4], Lumiere [2], WALT [8], and Zeroscope [42]. These models produce diverse left videos from a wide range of input text prompts, covering subjects such as humans, animals, buildings, and imaginary content.

Implementation Details.   To ensure the stereo effect appears realistic, we normalize the up-to-scale depth values predicted by the depth estimation model [46] to a range of (1, 10) and set the baseline between left and right views to 0.08. The frame matrix is constructed by evenly placing 8 cameras between the left and right views, with each camera corresponding to a warped video sequence. Due to the limitations of the zeroscope model, we currently conduct experiments on video sequences with 16 frames. Following the approach of RePaint [28], we employ DDPM [12] as our denoising scheduler with 1000 total time steps T and 50 denoising steps, resulting in 20 time step jumps per denoising step. During the initial 25 denoising steps (50 to 25), we resample 8 times at each step to establish a reasonable structure in disoccluded regions. For the remaining steps, we reduce resampling to 4 times and denoise only the right view for improved efficiency while generating stereoscopic videos. We run experiments on one A6000 GPU.

Baselines.   We compare our method with two families of approaches: video inpainting, and novel view synthesis from a monocular video. For video inpainting approaches, we generate the right view in the same manner as our method using depth-guided warping. We then apply state-of-the-art methods ProPainter [50] and E2FGVI [22] to inpaint the right views. For novel view synthesis methods, we compare our results with RoDynRF [27] and DynIBaR [24], which optimize scene representations relying on camera poses. To ensure a fair comparison, given the differing 3D scales between their reconstructed scenes and our estimated depth, we select the baseline for rendering the right view by matching the median disparity of foreground regions in the resulting disparity map to that of our methods. We are also aware of approaches trained on dedicated datasets that directly produce the right-view given the left-view like Deep3D [45]. However it does not generalize well to the generated video, especially those in non-realistic styles, and the comparison could be found in supplemental material.

4.1   Qualitative Results

Qualitative Comparisons. We show qualitative comparisons in Fig. 3. Previous video inpainting methods suffer from a common problem – the generated content in disoccluded regions is blurry, such as the knight's arm, horse's tail, and corgi's face, presumably because that these methods are trained on limited datasets. On the other hand, novel view synthesis methods suffer from unstable camera pose estimation (*e.g.*, DynIBaR fails on some videos). Though good at reconstructing visible content from the monocular video, they are typically poor at synthesizing novel contents in the disoccluded regions that are not observed in any frames (*e.g.*, ghost effect near the boundary in the RoDynRF result on the corgi example). In contrast, our approach takes advantage of generative capability of video diffusion models trained on massive scale datasets and does not require camera poses of the input video as inputs, thereby generating high-quality content in various types of scenarios (last row of Fig. 3) and consistently outperforms baseline methods. Additionally, we visualize the stereo

Figure 3: **Qualitative comparisons.** The first row shows left-view images. The video inpainting methods E2FGVI and ProPainter tend to generate blurry content in disoccluded regions, such as knight's arm and corgi's face. RoDynRF lacks the generation ability, thus content on the right side of the corgi case is poor. DynIBaR's results contain artifacts, and it requires camera poses as inputs, which failed in some scenarios. On the contrary, our method takes advantages of video generation models and is pose-free, thus generates high-quality content in different scenarios.

effects of different methods on the corgi case using a stereo depth estimator [21], which predicts disparity values from the stereo images. As shown in Fig. 12, RoDynRF and DynIBaR exhibit less depth variation, indicating weaker stereo effects. This occurs when the camera is wrong and training process overfits the training views, resulting in a sub-optimal 3D representation.

4.2 Quantitative Results

In this part, we show quantitative comparisons with other baselines. We primarily rely on a dedicatedly designed user study to evaluate the quality of generated stereoscopic video on various quality axes. We also provide an objective metric to measure the semantic similarity between the left and right views using pre-trained CLIP models.

Human Perception. To assess the perceived visual quality, we conducted a user study with 20 participants (9 female, age $\mu = 33, \sigma = 6.2$). On a VR headset, each participant viewed and evaluated five generated videos (out of 20 in total) by all five methods on stereo effect, temporal consistency, image quality, and overall experience using a 7-point Likert scale [25]. A total of 435 evaluations (DynIBaR failed to generate 13 videos) were counterbalanced and randomly shuffled. We also included a training session to eliminate novelty effects. Results are summarized in Table 3, with details in the supplemental material. Our method outperforms other baselines on measured metrics.

	E2FGVI	ProPainter	RoDynRF	DynIBaR	Ours
Stereo Effect ↑	4.79 (1.08)	4.81 (1.13)	2.97 (1.34)	1.86 (1.25)	**5.24** (0.94)
Temporal Consistency ↑	4.74 (1.33)	4.74 (1.22)	3.35 (1.66)	1.89 (1.33)	**5.15** (1.22)
Image Quality ↑	4.42 (1.27)	4.38 (1.28)	2.84 (1.60)	1.67 (1.07)	**5.12** (1.33)
Overall Experience ↑	4.67 (1.04)	4.66 (1.09)	2.92 (1.43)	1.72 (1.06)	**5.35** (0.99)

Table 1: **Quantitative comparisons.** This table reports results of human perception experiments as mean (std). Our method outperforms other baselines on all metrics. Kruskal-Wallis tests [18] reveal significant effects of group on all metrics ($\chi^2 > 13.3, p < 0.001***$). Post-hoc tests using Mann-Whitney tests [29] with Bonferroni correction reveal significant effects ($p < 0.05*, |r| > 0.1$) for each pairwise comparison, except E2FGVI *vs.* ProPainters yield comparable results.

Semantic Consistency. We additionally check the semantic consistency between the left and the right view. We use pre-trained CLIP model [35] to extract features for both left views and right views of a stereoscopic video, and then calculate the feature distance following Sun *et al.* [49] to obtain the semantic consistency score. In Table 2, our method attains the best semantic consistency (96.44) over other baselines.

4.3 Ablation Studies

Effect of Frame Matrix. In Fig. 4, we showcase that using frame matrix benefits semantic consistency between the left and right views. Without using frame matrix, the disoccluded regions in warped images can be inpainted with unconstrained contents, which are likely to be inconsistent with the left view given impressive generative capability of the diffusion model, such as the hair of the man and the head of the horse. This is also revealed in Table 2, where CLIP Score drops from 96.44 to 95.81 when disabling frame matrix. Thanks to constraints from other frames within the frame matrix, our method generates both reasonable foreground and background contents in the disoccluded regions. More studies of frame matrix are included in Sec.D of the supplemental material.

Effects of Disocclusion Boundary Re-Injection. In Fig. 5, we demonstrate the importance of updating unoccluded latent features for high-quality results. Without this update, the disoccluded region is inpainted with unnatural textures that don't blend well with the surrounding content. With the update, the inpainted content blends seamlessly. This is reflected quantitatively in Table 2, where the CLIP Score drops from 96.44 to 95.60 when unoccluded feature updates are discarded.

Method	E2FGVI	ProPainter	RoDynRF	DynIBaR	Ours - FM	Ours - DBR	Ours
CLIP ↑	94.34	95.29	96.03	93.24	95.81	95.60	**96.44**

Table 2: **Semantic consistency score**. We show the semantic consistency using CLIP feature similarity [10] between the left and right view. Our method outperforms previous methods as well as ablated cases.

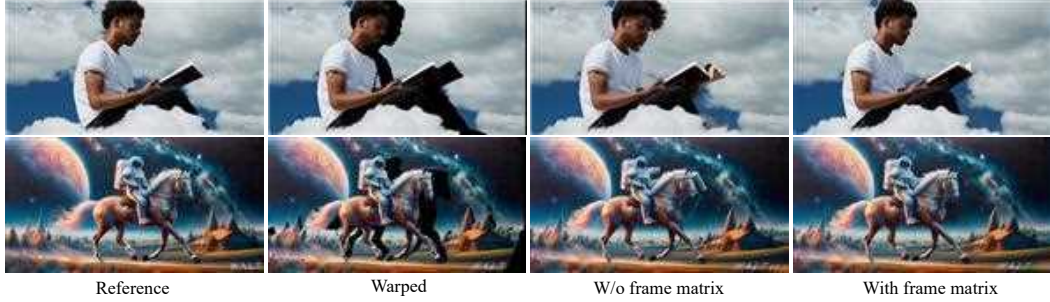| Reference | Warped | W/o frame matrix | With frame matrix |

Figure 4: **Semantically consistent content generation.** The reference frames are warped into the target view with disoccluded regions set to be black. Without using frame matrix, the generated content does not match the reference, such as the book and the face of horse. With frame matrix, the inpainted contents are more semantically reasonable.

| W/o Disocclusion Boundary Re-Injection | With Disocclusion Boundary Re-Injection |

Figure 5: **Disocclusion Boundary Re-injection.** Without disocclusion boundary re-injection, the inpainted images usually contain artifacts. Bottom-left corner shows the warped image.

Different Stereo Baselines. Fig. 6 shows increasing the stereo baseline makes inpainting harder and degrades stereoscopic video quality, as reflected by CLIP score. Our method is resilient to larger baselines, failing beyond 20cm (depth normalized to 1.0-10.0m). This range is sufficient for generating 3D stereoscopic video for most people, given typical inter-pupillary distances of 5-7cm.

| 7cm | 15cm | 23cm | 32cm | Semantic consistency |

Figure 6: **Result with different stereo baselines.** Unnatural artifacts begin to appear as the baseline expands. Our method performs well for stereoscopic video generation where baseline is usually less than 7cm.

5 Limitations

Although our results demonstrate the possibility of generating 3D stereoscopic videos using pre-trained video diffusion models, challenges remain. For one, we did not study longer videos because the architecture of a typical video diffusion model supports generating videos only a couple of seconds long. One possible solution for long 3D stereoscopic video generation is to use stronger foundational models, such as Sora [4]. Alternatively, we could gradually generate longer videos by overlapping frames of shorter videos. Additionally, our method is dependent on a depth estimation model [46], which may fail, *e.g.*, when dealing with thin structures.

6 Conclusion

We proposed a complete system for stereoscopic video generation, using a video diffusion model and our *frame matrix* inpainting scheme. Given the fast adoption of video generation, our approach bridges the gap between current ability to generate monocular and stereoscopic videos. In particular, we showed that our *frame matrix* formulation significantly advances the state-of-the-art for generative stereoscopic video, and can be adopted by existing and future video diffusion models.

9

References

[1] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Transactions on Graphics (TOG)*, 42(4):1–11, 2023.

[2] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Yuanzhen Li, Tomer Michaeli, et al. Lumiere: A space-time diffusion model for video generation. *arXiv preprint arXiv:2401.12945*, 2024.

[3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023.

[4] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024.

[5] Peng Dai, Yinda Zhang, Zhuwen Li, Shuaicheng Liu, and Bing Zeng. Neural point cloud rendering via multi-plane projection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7830–7839, 2020.

[6] Quankai Gao, Qiangeng Xu, Zhe Cao, Ben Mildenhall, Wenchao Ma, Le Chen, Danhang Tang, and Ulrich Neumann. Gaussianflow: Splatting gaussian dynamics for 4d content creation. *arXiv preprint arXiv:2403.12365*, 2024.

[7] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.

[8] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Li Fei-Fei, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. *arXiv preprint arXiv:2312.06662*, 2023.

[9] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. *Advances in Neural Information Processing Systems*, 35:27953–27965, 2022.

[10] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: a reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

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

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[13] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

[14] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7909–7920, 2023.

[15] 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. *arXiv preprint arXiv:2312.14937*, 2023.

[16] Animesh Karnewar, Andrea Vedaldi, David Novotny, and Niloy J Mitra. Holodiffusion: Training a 3d diffusion model using 2d images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18423–18433, 2023.

[17] 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.

[18] William H Kruskal and W Allen Wallis. Use of Ranks in One-criterion Variance Analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952.

[19] Yao-Chih Lee, Zhoutong Zhang, Kevin Blackburn-Matzen, Simon Niklaus, Jianming Zhang, Jia-Bin Huang, and Feng Liu. Fast view synthesis of casual videos. *arXiv preprint arXiv:2312.02135*, 2023.

[20] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022.

[21] Zhaoshuo Li, Xingtong Liu, Nathan Drenkow, Andy Ding, Francis X. Creighton, Russell H. Taylor, and Mathias Unberath. Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6197–6206, October 2021.

[22] Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17562–17571, 2022.

[23] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021.

[24] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4273–4284, 2023.

[25] Rensis Likert. A Technique for the Measurement of Attitudes. *Archives of Psychology*, 1932.

[26] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023.

[27] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13–23, 2023.

[28] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022.

[29] Henry B Mann and Donald R Whitney. On a Test of Whether One of Two Random Variables Is Stochastically Larger Than the Other. *The Annals of Mathematical Statistics*, pages 50–60, 1947.

[30] 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.

[31] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

[32] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021.

[33] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021.

[34] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[36] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[38] 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. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

[39] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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

[41] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[42] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023.

[43] Lezhong Wang, Jeppe Revall Frisvad, Mark Bo Jensen, and Siavash Arjomand Bigdeli. Stereodiffusion: Training-free stereo image generation using latent diffusion models, 2024.

[44] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7623–7633, 2023.

[45] Junyuan Xie, Ross Girshick, and Ali Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 842–857. Springer, 2016.

[46] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. *arXiv preprint arXiv:2401.10891*, 2024.

[47] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5336–5345, 2020.

[48] Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, et al. Wonderjourney: Going from anywhere to everywhere. *arXiv preprint arXiv:2312.03884*, 2023.

[49] SUN Zhengwentai. clip-score: CLIP Score for PyTorch. `https://github.com/taited/clip-score`, March 2023. Version 0.1.1.

[50] Shangchen Zhou, Chongyi Li, Kelvin CK Chan, and Chen Change Loy. Propainter: Improving propagation and transformer for video inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10477–10486, 2023.

[51] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

Appendix

In the supplementary sections, we provide more studies and details of the proposed method.

- In sec. A, we provide a pseudocode describing our frame matrix inpainting.
- In sec. B, it includes details of data preprocessing in handling warping-related artifacts.
- In sec. C, we include details of human perception experiments and provide additional comparisons with Deep3D.
- In sec. D, it contains more studies of frame matrix, including different trajectories in frame matrix and consistency across different views.
- In sec. E, we display more results in different scenarios.
- In sec. F, we show the effectiveness of our data preprocessing.

More video results and comparisons can be found in the supplementary webpage (.html).

A Algorithm Details

In the algorithm below, we present the detailed steps to denoise the Frame Matrix with spatial-temporal resampling, where we set $\mu_\theta(\mathbf{z}_t, c, t) = \frac{1}{\sqrt{1-\beta_t}}(\mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{z}_t, c, t))$, following DDPM[12].

Algorithm 1 Frame Matrix Inpainting

Input: $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$: Initial noisy latent maps
$\mathbf{z}_0$: Initial clean latent maps
for $t = T, ..., 1$ **do**
    for $n = 1, ..., N$ **do**
        if n is odd **then**
            Denoise time sequences $\{\mathbf{z}_{(s,:)t}|s = 1, ..., S\}$:
            for $s = 0, .., S$ **do**
                $\mathbf{z}_{(s,:)t-1}^{\text{known}} \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{z}_{(s,:)0}, (1 - \bar{\alpha}_t)\mathbf{I})$
                $\mathbf{z}_{(s,:)t-1}^{\text{unknown}} \sim \mathcal{N}(\mu_\theta(\mathbf{z}_{(s,:)t}, c, t), \Sigma_\theta(\mathbf{z}_{(s,:)t}, c, t))$
                $\mathbf{z}_{(s,:)t-1} = \mathbf{m}_{(s,:)} \odot \mathbf{z}_{(s,:)t-1}^{\text{known}} + (1 - m_{(s,:)}) \odot \mathbf{z}_{(s,:)t-1}^{\text{unknown}}$
            end for
        else
            Denoise view sequences $\{\mathbf{z}_{(:,v)t}|v = 1, ..., V\}$:
            for $v = 0, .., V$ **do**
                $\mathbf{z}_{(:,v)t-1}^{\text{known}} \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{z}_{(:,v)0}, (1 - \bar{\alpha}_t)\mathbf{I})$
                $\mathbf{z}_{(:,v)t-1}^{\text{unknown}} \sim \mathcal{N}(\mu_\theta(\mathbf{z}_{(:,v)t}, c, t), \Sigma_\theta(\mathbf{z}_{(:,v)t}, c, t))$
                $\mathbf{z}_{(:,v)t-1} = \mathbf{m}_{(:,v)} \odot \mathbf{z}_{(:,v)t-1}^{\text{known}} + (1 - m_{(:,v)}) \odot \mathbf{z}_{(:,v)t-1}^{\text{unknown}}$
            end for
        end if
        Add back one noise step for resampling:
        $\mathbf{z}_t \sim \mathcal{N}(\sqrt{1 - \beta_{t-1}}\mathbf{z}_{t-1}, \beta_{t-1}\mathbf{I})$
    end for
end for

B Details of Data Preprocessing

Multi-Plane projection. Given RGB-D images, we warp them into a target camera view. Instead of projecting all pixels onto one image plane and handling occlusions using z-buffer, we divide the camera view space into multi-plane images $\{I_1^{step0}, ..., I_N^{step0}\}$ (N=4 in this paper) according to near and far depths, then each pixel is projected onto the image plane closest to it. We use $\{M_1^{step0}, ..., M_N^{step0}\}$ to indicate valid pixel positions on each image plane. By doing this, the foreground and background are separated in different planes temporarily, which makes dealing with artifacts (i.e., isolated points and entangled foreground and background content in Fig. 11 left) easier.

Remove isolated points. Due to the inaccuracy of depth values around image boundaries, these pixels are warped into wrong positions leading to isolated pixels (see red box in Fig. 11 left). Intuitively, isolated pixels

Figure 7: **Videos in frame matrix.** In both cases, each column is a generated video in a camera, and each row represents generated frames in different cameras at a specific timestamp.

have no or very few neighbors, thus we detect isolated pixels based on this observation. Specifically, we apply convolution on each mask plane M_i^{step0} using a 3×3 kernel, after which isolated pixels are empirically determined where values after convolution are less than 0.5. We remove these isolated pixels on both RGB and mask planes to obtain new $\{I_1^{step1}, ..., I_N^{step1}\}$ and $\{M_1^{step1}, ..., M_N^{step1}\}$.

Handle foreground and background entanglement. Since the depth image is not a watertight representation, the warped image usually contains small cracks/holes that confuse foreground and background content. For example, the gray road can be seen through the dog's ear in Fig. 11 left. Similar to handling isolated pixels, we use a 3×3 Gaussian kernel to perform convolution on each mask plane M_i^{step1}. When there are cracks, the values after convolution will be less than 1. In this paper, positions with no pixel values (0 in M_i) but with greater values than 0.2 after convolution are considered cracks. We fill these cracks via interpolating nearby valid pixels in each image plane and obtain new multi-plane images $\{I_1^{step2}, ..., I_N^{step2}\}$ and $\{M_1^{step2}, ..., M_N^{step2}\}$.

After handling artifacts in each image plane, all image planes are blended into one image (e.g., Fig. 11 ours left) in a back-to-front order using Eq. 11, where the content of front plane blocks content belongings to the plane at the back.

Figure 8: **Videos in frame matrix constructed using a spiral trajectory.** Warped and generated frames in different cameras at different timestamps.

Figure 9: **Consistency.** The content is inconsistent when each view is generated independently. Frame matrix benefits the consistency of our results across different views. Please note the dragon's wing.
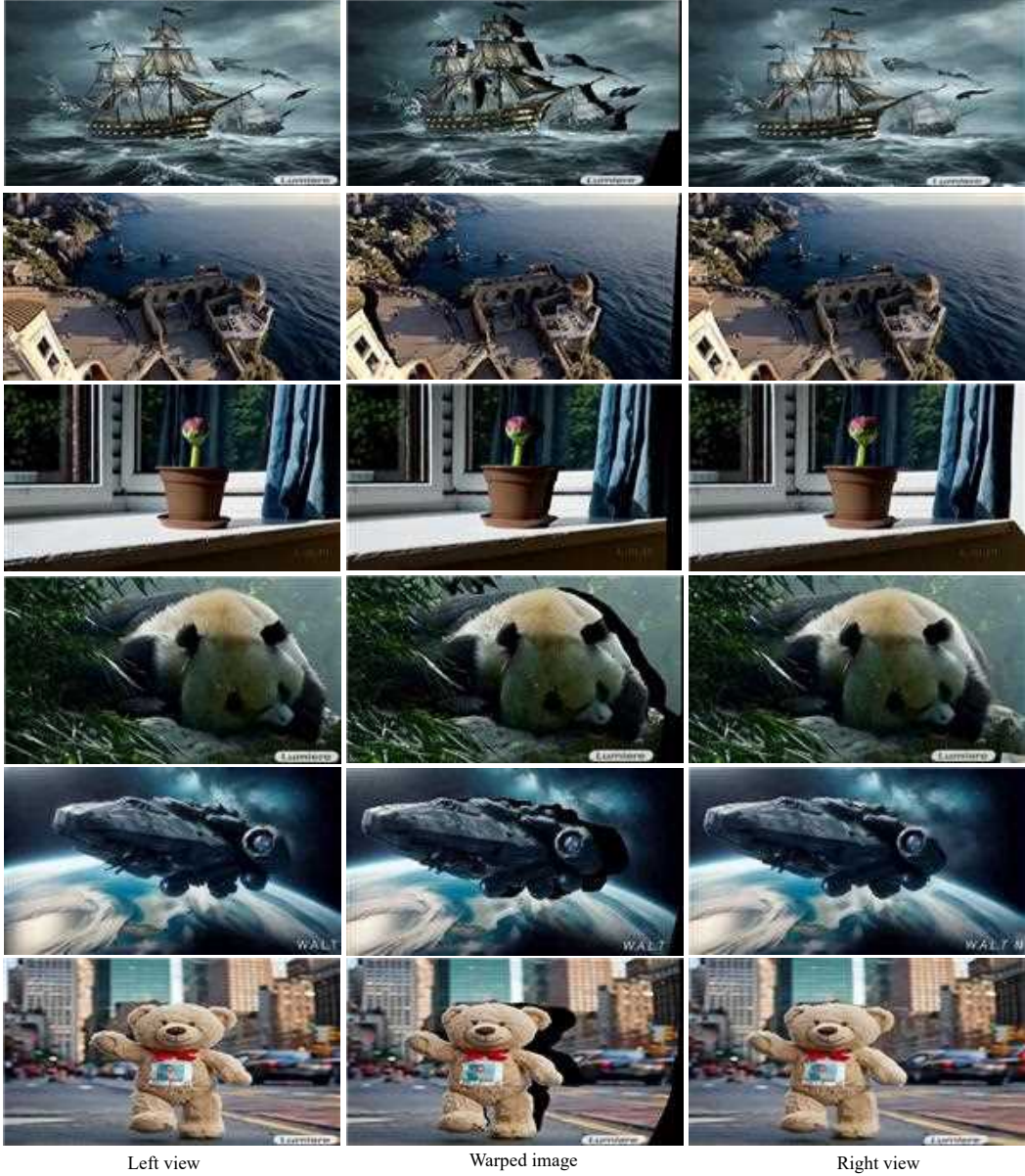
| Left view | Warped image | Right view |

Figure 10: **More results.** We display more generated results in different scenarios.
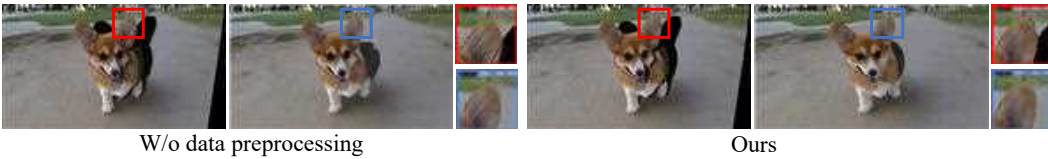

| W/o data preprocessing | | Ours |

Figure 11: **Data preprocessing.** Left: without handling isolated points and entangled foreground and background (the gray road can be seen through the dog's ear) in warped images, these artifacts remain in the final results. Right: our results have no artifacts.

$$I = I \times (1 - M_i^{step2}) + I_i^{step2} \times M_i^{step2}, \; for \; i \; in \; [N, ..., 1]. \tag{11}$$

| E2FGVI | ProPainter | RoDynRF | DynIBaR | Ours |

Figure 12: **Disparities.** We visualize stereo effects by predicting disparity values from stereo images [21].

Left view   Right view

| t | t+1 | t (warped) | t (inpainted) |

Figure 13: **Ability to utilize unobserved content.** Left view: two consecutive images observed by the left view. Right view: the warped and inpainted images at time t. Note that the black region is inpainted with the character "R", matching the characters in the second image at time t+1.

C Details of Human Perception Study

Participants. To evaluate the perceived quality of the generated stereoscopic videos, we recruited 20 participants (9 females) at least 18 years old ($\mu = 33, \sigma = 6.2$) with normal or corrected-to-normal vision at an anonymous institution via email lists and group communication software. The majority of participants had some experience with virtual reality. None of the participants was involved with this project prior to the user study.

Study setup. The study was conducted in a quiet meeting room with a commercial VR headset as the primary apparatus. The study software is implemented in Unity 2023.3.0b and we render stereoscopic videos with custom shaders on a $1.8m \times 1.0m$ quad that is three meters away from the participant in the world space, which occupies approximately 33.4 degrees in width and 18.92 degrees in height initially. Users have the freedom to move themselves within the meeting room to examine the stereoscopic video. This setup allowed participants to experience the stereoscopic videos in virtual reality settings and provided a controlled environment for the user study.

Study protocol. Each study session consists of a demographics interview with consent forms, a training session, and an evaluation session. To eliminate the ordering effect, we randomly counterbalanced all five methods for each video and assigned five random videos (out of 20 videos) with five conditions to each participant. However, since DynIBaR method failed to generate 13 videos, we collected a total of $5 \times 5 \times 20 - 13 \times 5 = 435$ evaluations from 20 participants, resulting in 100 human evaluations for each method except DynIBaR. During the training session, we randomly picked a video that was outside of the assigned videos to the participant and asked the participant to rate the stereoscopic effect, temporal consistency, graphical quality, and overall experience on a 7-point Likert scale [25], with 1 being the lowest, 7 being the highest, and 4 being the average. This procedure helps eliminate the novelty effect and calibrate the user's rating before the formal evaluation session. In the formal evaluation, we prompted the participant with the question like *"How would you like to rate the stereoscopic effect of the video on a 7-point scale, with 1 being the lowest, 7 being the highest, and 4 being the average?"* and asked the user the reason behind the rating.

Metrics.. We evaluate the perceived quality of generated stereo videos based on three key aspects: 1. Stereo Effect. This refers to the perception of depth achieved by presenting slightly different images to each eye. A strong stereo effect makes objects appear closer or farther away, enhancing the 3D experience. Example questions: "How strong was the 3D effect in the video?" and "Which video felt more immersive due to the 3D effect?" 2. Temporal Consistency. This aspect assesses the smoothness of scene motion and the absence

| Left view | Right view | Disparity |

Figure 14: **Results of Deep3D.** Deep3D does not provide the function to change the stereo baseline, and the vague disparity map on the right side demonstrates its weak stereo effects.

of artifacts such as jitter or ghosting over time. Example questions: "How smooth and natural did the motion of objects appear?" and "Did you notice any flickering, jumpiness, or distortions in the video?" 3. Graphical Quality. This evaluates the overall visual appeal of the video, including the quality of details, textures, lighting, and color fidelity. Example questions: "How would you rate the visual quality of the video?" and "Which video had more detailed and realistic textures?"

Study results. Overall, despite the missing data points for the DynIBaR method in some videos, Kruskal-Wallis tests [18] reveals significant effects of group on all metrics respectively ($\chi^2 > 13.3, p < 0.01$): with stereo effect $\chi^2 = 186.3, p < 0.001$, temporal consistency $\chi^2 = 121.3, p < 0.001$, graphical quality $\chi^2 = 153.2, p < 0.001$, and overall experience $\chi^2 = 192.9, p < 0.001$. We further performed post-hoc tests using Mann-Whitney tests [29] with Bonferroni correction, which revealed significant effects ($p < 0.05, |r| > 0.1$) for each pairwise comparison, except E2FGVI *vs.* ProPainters. Specifically, for Ours *vs.* E2FGVI, $p = 0.002$ on stereo effect, $p = 0.030$ on temporal consistency, $p < 0.001$ on graphical quality and overall experience. For Ours *vs.* ProPainter, $p = 0.004$ on stereo effect, $p = 0.017$ on temporal consistency, $p < 0.001$ on graphical quality and overall experience.

Study findings. Our results suggest that our methods achieve significantly better perceived stereoscopic effect than all other methods, while improvement in graphical quality and overall experience is more evident over stereoscopic effect; and stereo effect more evident over temporal consistency. During the study, we also observed many positive comments about our methods like "the contour is more clear", *"the graphics are sharper with fewer artifacts"*; however, we also observed negative or neutral feedback like *"some part really works and some parts don't: one side of the turtle face is wrong"*, and *"I see no difference (on the faces)"* from two participants. This suggests future research to investigate holistic perceptual consistency in stereoscopic videos and finetune models for special subjects like human beings.

Additional User Study on Ours *vs.* Deep3D.

Despite that we did not include Deep3D in the design of our initial user study, we further conducted a human evaluation between Ours and Deep3D across the same metrics with a total of 190 random evaluations over 20 random videos, following the same protocol. Pairwise Mann-Whitney tests with Bonferroni correction reveal significant effects on stereo effect ($p < 0.001$), overall experience ($p < 0.001$), and temporal consistency ($p = 0.015$). We found our method outperforms Deep3D in stereo effect and overall experience, yet falling slightly short in temporal consistency.

Similar to Fig.4 in main paper, we visualize Deep3D's disparity map in Fig. 14. The vague disparity map in the third column demonstrates weak stereo effects, which matches the statistic results in Table 3. By manually modifying the disparity map or changing the stereo baseline, 3D effects may become apparent. However, Deep3D does not support these functions.

	Deep3D	Ours	p-value	$|r|$ (effect size)
Stereo Effect ↑	2.29 (1.63)	**5.29** (1.09)	< 0.001 ***	0.60 (large)
Temporal Consistency ↑	**5.37** (1.23)	5.06 (1.25)	0.015 *	0.49 (medium)
Image Quality ↑	**5.27** (1.17)	5.12 (1.19)	0.103	0.10 (small)
Overall Experience ↑	3.68 (1.36)	**5.08** (1.09)	< 0.001 ***	0.57 (large)

Table 3: **Quantitative comparisons.** This table reports results of human perception experiments as mean (std) between Deeph3D and Ours. Our method outperforms Deeph3D in stereo effect and overall experience, yet falls slightly short in temporal consistency. Mann-Whitney tests with Bonferroni correction reveals significant effects on stereo effect ($p < 0.001, Z = -8.24$), overall experience ($p < 0.001, Z = -7.92$), and temporal consistency ($p = 0.015, Z = -6.72$).

D   More Results of Frame Matrix

Other trajectories in frame matrix. In main paper, we show generated 3D left and right views. Here, we additionally show the results of other trajectories. In Fig. 7, we selectively display frames generated within the frame matrix at different timestamps (3 out of 16) in different camera views (3 out of 8). From the results, both foreground and background content are coherent across different frames. Moreover, instead of constructing frame matrix using cameras moving from left to right, we alternatively move the camera following a spiral trajectory. In Fig. 8 first and third rows, we selectively show the warped images in different camera views (3 out of 16), where disocclusions appear around the plane. Under each warped image, we display the corresponding image with disocclusions filled.

Consistency. In Fig. 9, the first row is warped images under different camera views. We generate each view independently and show results in the second row, where the content is not consistent across different views,

such as the dragon's wing. With the help of the frame matrix, which also regularizes generation in the direction of camera motion, our results in the third row are more consistent.

E   More Results of Stereoscopic Videos

More cases. In this part, more generated results are displayed in Fig. 10. The proposed method works on different scenarios, such as the beautiful church, imaginary scenes, and ships in the storm where the whole scene is dynamic. The high-quality generated results in Fig. 10 right column demonstrate the generalization ability of the proposed method.

Ability to utilize temporal context for inpainting. Our method is able to harmonize image contents between different temporal frames during inpainting and thus enhance temporal consistency. Figure 13 shows one example. When inpainting the right-view frame at t, our method successfully creates content that is consistent with the left-view frame at $t + 1$ (see the generated character "R" in the disoccluded region). Note that such consistency is maintained automatically thanks to frame matrix based denoising, since all temporal frames are taken into account.

F   More Ablation Studies

Effects of Data Preprocessing. In Fig. 11 left, obvious artifacts are in warped images, such as isolated points and cracks where the foreground ear is entangled with the background gray road, and these artifacts remain in the final generated results. On the contrary, Fig. 11 right shows our results, which are artifacts-free after applying the proposed data preprocessing.