Independent Study - Stereo Video Generation

Dev Gupta
Washington University in St. Louis
{dev.g}@wustl.edu

Figure 1: Left and generated right view video freeze frame with the prompt: "A mystical unicorn galloping over a rainbow with sparkles shimmering in its wake".

1. Problem Statement

Given an arbitrary video, we define this as a left-view. We may be provided a ground-truth disparity map parameterized by depth Z, focal length f, and baseline T. Otherwise, we are only given f and T, and we will estimate depth Z using a pre-trained depth estimation model to obtain disparity d. Our task is to generate a geometrically, spatially, and temporally consistent right-view video corresponding to this left-view video. Given timestep t for each left-view frame I, a respective frame image $\hat{I}$ will be generated.

2. Goals

At a minimum, I wish to explore the video generation pipeline and experiment with stereo generation. At a maximum, I wish to succeed in generation and attempt cycle consistency with focus on improving output.

Detailed/Enumerated goals are as follows:

1. Research existing methodology on video generation.
2. Explore stereo generation extensions on various domains.
3. Successfully implement the Wan2.1 pipeline on RIS compute resources.
4. Verify Video-Depth-Anything results on natural/synthetic inputs.
5. Integrate both key repositories into a single pipeline.
6. Perform depth interpolation and apply novel latent warping consistent across the temporal domain.
7. Finetune inference-time hyperparamters in order to succeed in stereo video generation.
8. Explore Null Inversion for video latents.
9. Apply warp from right to left rather than left to right to generate new loss function. (non-training-free)

3. Methodology

There are multiple approaches and techniques to accomplish this task. Two broad categories encompassing these approaches are training and training-free. For the purposes of minimizing compute needed for such a high-dimensional domain, we plan to take the latter.

Specifically, we wish to incorporate two foundational ideas for this task. First, perform latent manipulation in order to guide diffusion for the generated right-hand view through a means of warping. Second, perform left-right cycle consistency by warping the generated right-view and verifying the left-view is consistent with the right geometrically. The first idea will be the backbone of our methodology, and the second will be implemented in hopes of improving fidelity.

4. External Guidance

The biggest contributor to help formulate my work was from Feng. Additional help was received by Brian and Eric. Feng provided me with various resources as well as assistance with code (as he has worked on stereo generation himself through GenStereo). He provided me with access to the TartanAir dataset initially when training-free was not a consideration. He introduced various new concepts in order to solve the problem statement such as 4D gaussian splatting manipulation and cycle consistency.

5. Data

Initally, the TartanAir was pre-downloaded and stored by Feng for my use. It consisted of various left/right views of a synthetic scene coupled with accurate ground truth depth, perfect for this problem statement. In the end through discussion, we decided it was best to focus on training-free due to the lack of industry-scale compute which other papers in the field had. This is simply the nature of the video domain.

6. Progress

I have made substantial progress in the entire pipeline. Most of my enumerated goals were met. This includes performing in-depth research and advancing my knowledge from 0 to 1. I gained insight into the various methods used in this task. Then, I was able to apply my knowledge to start developing. I instantiated the pipeline for each individual part. This includes the Wan2.1 and Video-Depth-Anything. One key hurdle was handling package etiquette and managing inter-dependencies between these repositories (especially with flash attention).

Then, once these two pipelines were developed, I integrated both into a custom text-to-video class that extends the original. After much debugging, I was able to create a depth map of a newly generated video within the pipeline. Then, I converted the depth into disparity and set default parameters such as a scale factor of 8. In order to shift, I needed to interpolate the depth map first through trilinear interpolation to a shape of $T * H * W$. After applying and setting empty pixels to default 0.0, I generated a boolean mask in order to later apply that shift in further time-steps. The stereo pixel shift, or in other words, warp, took in a left view (latent) of shape $B * C * T * H * W$ alongside a depth map of $T * H * W$, and outputted a mask of $C * T * H * W$ and a warped right view of $B * C * T * H * W$.

After successfully doing so and verifying results, I passed it through a shift schedule of every 10 time-steps. This proved to be unsuccessful, even though it worked for StableDiffusion on imagery. Wan2.1 uses UniPC/DPM which is a dedicated high-order solver for diffusion ODEs. This is contrary to StableDiffusion used in the Training-Free Stereo Diffusion paper which is first-order, or markovian. This is important to note because the right view latent will maintain momentum from the left view, hindering its ability to generate coherent right views. These right views will be heavily weighted by the left and not incorporate the warp meaningfully. Results showed a right view that was very similar to the left with minor changes. The changes did not include a disparity map shift due to the high momentum from the left view from the schedulers. To mitigate this, I set the shift to every timestep, especially starting from step 0. By doing this, the momentum was lost and the shift was present. I will go into the limitations now of this.

7. Current Limitations and Results

Current limitations are apparent in the results. There exists both a left view and right view merged into one. This is due to various potential reasons. One is that, I don't currently, but I could and correctly should inject a timestep and prompt once for both views in a batch. This can cause a high dependence on the left view towards the right due to the shared resources. Duplicating them will cause more independence as wanted, but results in deep divergence. Another is due to the momentum of the scheduler. Ideally, a DDPM scheduler is available through Wan, but currently isn't.

There is a clear effect of the stereo pixel shift that has been extended successfully to the video domain. I was hesitant about interpolating across the temporal domain, but the reshaped depth map was ideal for shifting/warping the left view correctly. Adjusting the scale factor also gave ideal results in the right view (even though the generated video included the left view as well). In addition, shifting the latents across their "percieved" spatial dimensions within the latents proved that the latents dimension preserved the information within that dimension. Temporal axis was unaffected, as intended. This showed that the shift could be and is successful.

8. Future Steps

Find the harmonic balance between including the duplicated timestep and prompt, and the number of time-steps that the warp is overlayed onto the right view. That is my immediate next step. If it is successful, then it is worth keeping. Otherwise, I must research additional solutions to this issue. This might include trying a new approach I have thought about. This approach is to pass the left view throught he VAE encoder and repeatedly warp and apply that warped view for further end-goal consistency. Other than that, I will continue through my enumerated goals mentioned previously. [1] [2] [3]

References

[1] Shengnan Zhu Feihu Zhang Zilong Huang Jiashi Feng Bingyi Kang Sili Chen, Hengkai Guo. Video depth anything: Consistent depth estimation for super-long videos. *arXiv preprint*, 2024. 2

[2] Lezhong Wang, Jeppe Revall Frisvad, Mark Bo Jensen, and Siavash Arjomand Bigdeli. Stereodiffusion: Training-free stereo image generation using latent diffusion models. *arXiv preprint*, 2024. 2

[3] WanTeam. Wan: Open and advanced large-scale video generative models. *arXiv preprint*, 2024. 2