GS-DiT: Advancing Video Generation with Dynamic 3D Gaussian Fields through Efficient Dense 3D Point Tracking – Supplementary

Weikang Bian[1,2] Zhaoyang Huang[3] Xiaoyu Shi[1]
Yijin Li[3] Fu-Yun Wang[1] Hongsheng Li[1,2]

[1]CUHK MMLab [2]CPII under InnoHK [3]Avolution AI

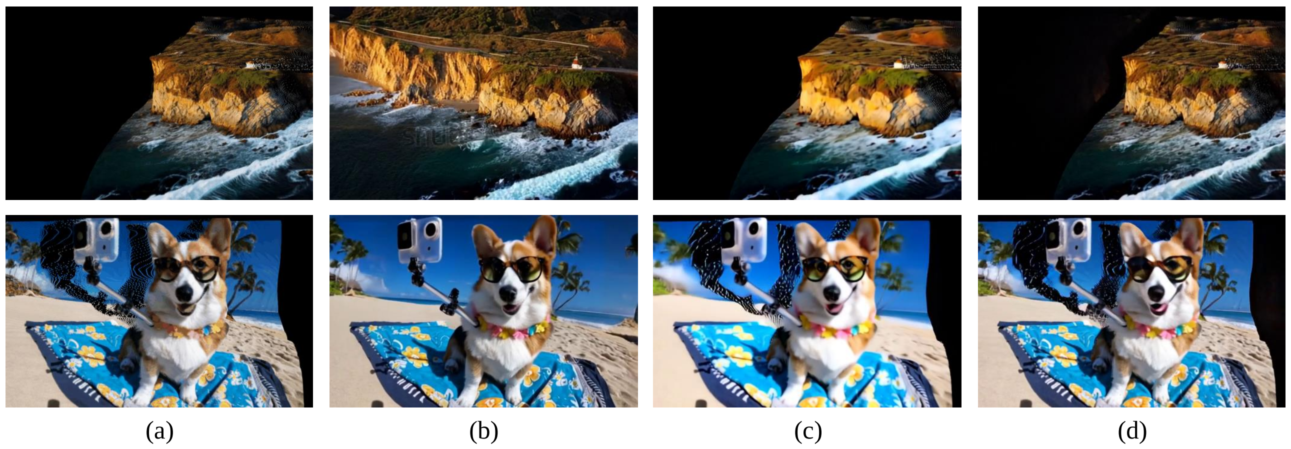
Figure 1. Comparison with video inpainting. (a) is the input frame. (b) is the video frame generated by our GS-DiT. (c) and (d) are the video frames generated by Inpainting-A and Inpainting-B.

Methods	DAVIS AJ ↑	DAVIS $< \delta^x_{avg}$ ↑	DAVIS OA ↑
RGB-RAFT (DOT) [4]	60.1	74.5	89.0
RGBD-RAFT	55.7	71.6	86.7
D3D-PT (Ours)	**63.4**	**75.2**	**89.4**

Table 1. Ablation Study on DAVIS.

1. Ablation Study

Dense 3D Point Tracking. We conduct an ablation study on DAVIS to show the superiority of our loosely coupled dense 3D point tracking design. As shown in Tab. 1, DOT [4] can be regarded as the baseline that takes the original RAFT to estimate the dense 2D point tracking. Directly extending the RAFT to accept RGB-D images as inputs (RGBD-RAFT) degrades the point tracking accuracy seriously because the depth distribution in training is different from the distribution of depth estimated in the inference stage. On the contrary, our loosely coupled 3D point tracking design D3D-PT improves the tracking accuracy on the DAVIS.

GS-DiT GS-DiT generates video conditioned on the input video, which is rendered from the dynamic 3D Gaussian field. The generated video is expected to fix the artifacts, such as the blurs and the incomplete areas, derived from the imperfect dynamic 3D Gaussian field. Such a process is similar to video inpainting, so we set a simple baseline that trains a DiT-based video inpainting model to reveal the essence of building the dynamic 3D Gaussian fields for training. We corrupted the videos with two simple masks: evenly distributed dispersed masks (Inpainting-A) and fixed-size grid masks at random locations (Inpainting-B). We train all models with 5000 iterations at 320×512 resolution. We expect that the masked region occupies 40% of the images, so we set the dispersed occlusion ratio as 40% in Inpainting-A and the grid mask with the size of 256×256 in Inpainting-B. We remove part of the information to obtain the corrupted video according to the generated random mask as the condition video. As shown in Fig. 1, our GS-DiT obtains clear details and infers reasonable unobserved regions. In contrast, both inpainting models (Inpainting-A and Inpainting-B) fail to infer the incomplete regions and present blurry video frames. Moreover, the blurry effects is severe in the video frame generated by Inpainting-A. This comparison shows that building the dynamic 3D Gaussian fields with our D3D-PT is the cornerstone of the GS-DiT. We provide more demonstrations in

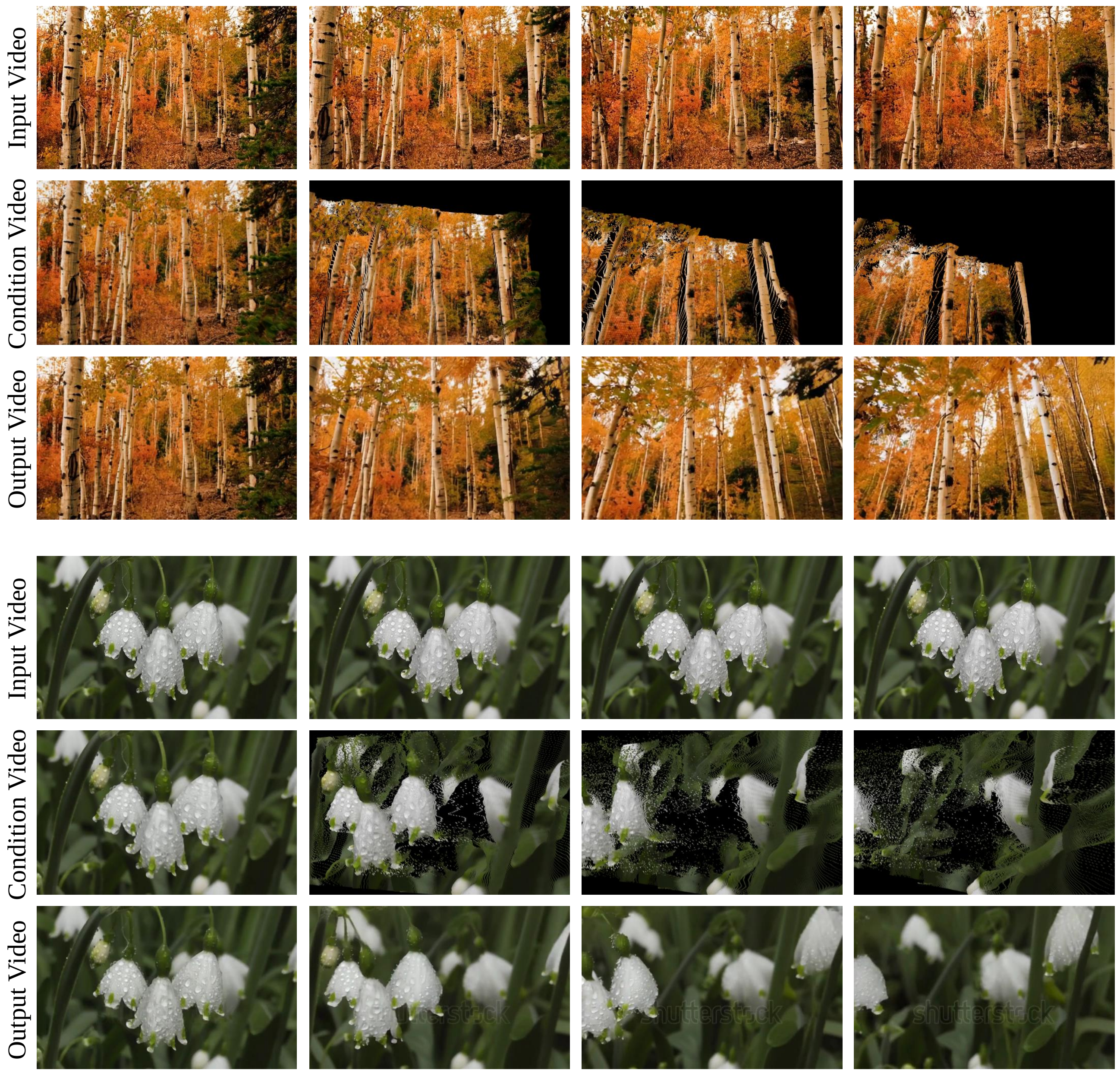

Figure 2. Demonstrations of GS-DiT. GS-DiT generate the Output Video from the Condition Video, which is rendered from the Input Video according to the camera pose trajectory.

Methods	Subject Consistency ↑	Background Consistency ↑	Motion Smoothness ↑	Aesthetic Quality ↑	Imaging Quality ↑	Temporal Flickering ↑
MonST3R	83.84%	89.61%	94.63%	44.07%	**64.59%**	88.13%
GCD	87.79%	91.26%	97.79%	50.76%	59.75%	94.20%
GS-DiT	**93.72%**	**94.18%**	**98.69%**	**52.32%**	62.76%	**94.87%**

Table 2. Quantitative comparison on VBench [2].

Fig. 2. Although the rendered videos are highly corrupted, the output videos generated by our GS-DiT are still of high quality.

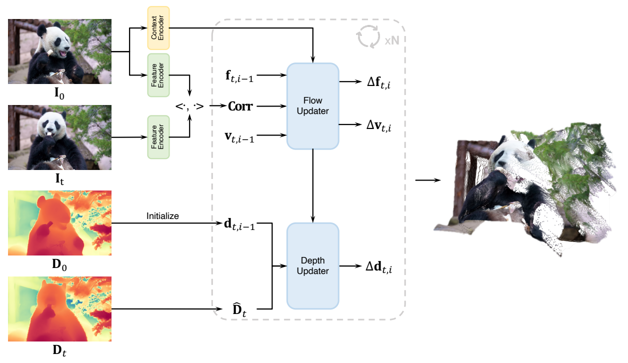

Figure 3. **The neural network architecture of D3D-PT.** D3D-PT predicts dense 3D point tracking for an input RGB-D video.

2. Qualitative Comparison on VBench

We compare our GS-DiT with MonST3R and GCD through the VBench [2] metrics in Tab. 2. GS-DiT demonstrates superior performance across five metrics, achieving the highest scores in subject consistency, background consistency, motion smoothness, aesthetic quality, and temporal flickering reduction. This comprehensive dominance suggests GS-DiT's robust capability in maintaining spatio-temporal coherence while delivering visually appealing content. MonST3R presents unreasonable high imaging quality score because the MUSIQ [3] used for imaging quality score prediction is not trained on corrupted images.

3. More Details about D3D-PT

RAFT [5] is designed for optical flow estimation, *i.e.*, regressing a per-pixel displacement field $\mathbf{f}_t : \mathbb{I}^{H\times W\times 2} \rightarrow \mathbb{R}^{H\times W\times 2}$ that maps each source pixel to the coordinate $\mathbf{x}_t$ in the target video frame t. Suppose the coordinate of the pixels in the source image is $\mathbf{x}_0$, the target coordinate can be derived from $\mathbf{x}_t = \mathbf{x}_0 + \mathbf{f}_t$. DOT [4] adopt RAFT to refine the dense point tracking. Drawing inspiration from RAFT and DOT, we design the network architecture of our D3D-PT by iteratively refining the dense 3D point tracking, including the 2D point tracking $\mathbf{x}_t$, the visibility $\mathbf{v}_t$, and the depth $\mathbf{d}_t$, as shown in Fig. 3. We refine the 2D point tracking $\mathbf{x}_t$ and the visibility $\mathbf{v}_t$ through a flow updater, following RAFT, and refine the depth $\mathbf{d}_t$ with a depth updater that is loosely coupled to the flow updater.

RAFT encodes image features with shallow CNNs, computes correlation volumes for all pairs of pixel features, builds 4-layer correlation pyramids by average pooling, and iteratively refines the correspondence estimation with a recurrent decoder according to the correlation pyramids and the image features. Following RAFT, given a pair of RGB images, we encode them with a siamese network as $\mathbf{feat}_0 \in \mathbb{R}^{H\times W\times C}$ and $\mathbf{feat}_t \in \mathbb{R}^{H\times W\times C}$ corresponding to the source image and the target image. H, W, C denote the height, width, and channels of the encoded feature map.

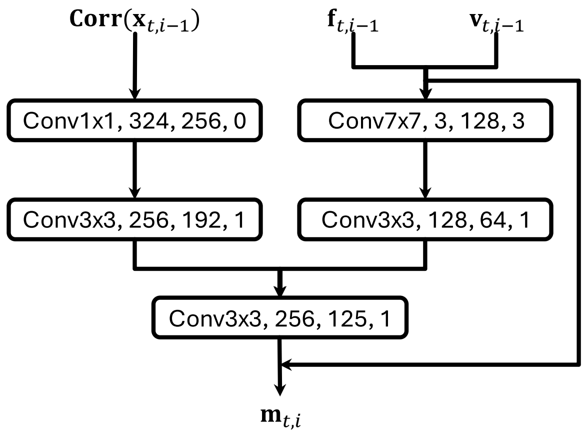

Figure 4. The structure of $MotionEnc_{flow}$.

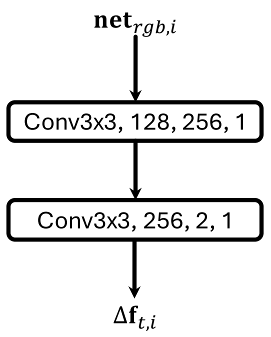

Figure 5. The structure of $FlowHead$.

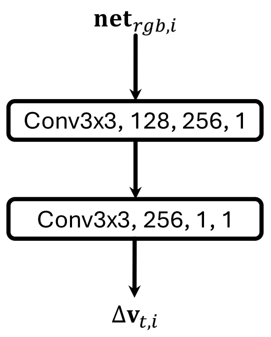

Figure 6. The structure of $VisHead$.

The source image is additionally encoded with a CNN to provide the context information $\mathbf{inp} \in \mathbb{R}^{H\times W\times C/2}$. With the correlation pyramid $\mathbf{Corr}$ built from $\mathbf{feat}_0$ and $\mathbf{feat}_t$, we predict the residual flow and visibility via the flow updater, including a motion encoder $MotionEnc_{flow}$, a recurrent unit $ConvGRU_{flow}$, a flow prediction head $FlowHead$, and visibility prediction head $VisHead$. We encode the motion fea-

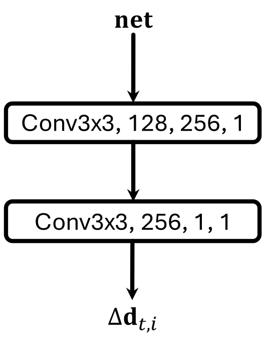

Figure 7. The structure of *DepthHead*.

ture $\mathbf{m}_{t,i}$ from the flow and visibility estimated at the last iteration $\mathbf{f}_{t,i-1}, \mathbf{v}_{t,i-1}$, and cropped correlation information $\mathbf{Corr}(\mathbf{x}_{t,i-1})$:

$$\mathbf{m}_{t,i} = MotionEnc_{flow}(\mathbf{f}_{t,i-1}, \mathbf{Corr}(\mathbf{x}_{t,i-1}), \mathbf{v}_{t,i-1}). \quad (1)$$

We show the structure of the motion encoder in Fig. 4. There is a ReLU [1] activation function between the convolution layers. The motion features $\mathbf{m}_{t,i}$ will be fed to the $ConvGRU_{flow}$ to estimate the flow update:

$$\begin{aligned} \mathbf{net}_{rgb,i} &= ConvGRU_{flow}(\mathbf{m}_{t,i}, \mathbf{net}_{rgb,i-1}, \mathbf{inp}), \\ \Delta\mathbf{f}_{t,i} &= FlowHead(\mathbf{net}_{rgb,i}), \\ \Delta\mathbf{v}_{t,i} &= VisHead(\mathbf{net}_{rgb,i}). \end{aligned} \quad (2)$$

$MotionEnc_{flow}$, $ConvGRU_{flow}$, and $FlowHead$ are standard blocks used in DOT. $\mathbf{net}_{rgb}$ is an iterativelly updated hidden state. $ConvGRU_{flow}$ is the recurrent decoder used by RAFT. $FlowHead$, $VisHead$, and $DepthHead$ used to regress residual flow, visibility, and depth share similar structures. We show them in Fig. 5, Fig. 6, and Fig. 7. Besides updating flow estimation, the motion feature $\mathbf{m}_{t,i}$ and the hidden feature $\mathbf{net}_{rgb,i-1}$ will also be used in the depth refinement as elaborated in the main paper. We assign $\mathbf{net}_{rgb,i-1}$ to $\mathbf{net}$ used in the main paper.

References

[1] AF Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018. 4

[2] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024. 2, 3

[3] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5148–5157, 2021. 3

[4] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Dense optical tracking: Connecting the dots. In *CVPR*, 2024. 1, 3

[5] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 3