FastEventDGS: Deformable Gaussian Splatting for Fast Dynamic Scenes from a Single Event Camera

Supplementary Material

1. Discussion

Difference with simple motion constraint. While existing motion constraints assume nearby Gaussians have similar deformations, minimizing these differences provides only weak supervision and fails during large deformations or complex object contacts. In contrast, our novel motion loss (L_m) provides direct supervision derived explicitly from the temporal nature of the event stream itself, ensuring that the learned deformation remains physically consistent with the actual captured motion. Additionally, our method is more time-efficient as it eliminates the need for a k-nearest neighbor graph (note that Gaussian positions change during optimization, which requires iterative reconstruction of the graph).

Usage of VGGT. Our approach consists of sending a single rendered image to VGGT, and this significantly enhances reconstruction quality. However, we would like to clarify that our framework is "event-only" by definition, as it requires only event data as input. Furthermore, by sending in a single frame, we do not use VGGT's reconstruction abilities, but merely leverage a geometric prior from a large-scale pre-trained network to stabilize the optimization. This could perhaps be exchanged versus a depth prior obtained from events, but the lack of absolute intensities makes direct depth prediction an ill-posed problem. It is important to note that the final fidelity of geometry and–especially–deformation are driven entirely by the event stream's temporal gradients, which VGGT cannot provide.

2. Extended Experiment

2.1. Comparison with RGB Camera

To further evaluate the advantages of event-based sensing, we conducted a controlled synthetic experiment using Blender to simulate high-speed motion. As shown in Table 1 (*BlurFrameDGS*), we compared our method against RGB-based reconstruction at a low temporal resolution of 10 fps. The results demonstrate that while RGB frames suffer from severe motion blur and aliasing under these conditions, our event-only approach maintains high-fidelity reconstruction, empirically verifying the robustness of event-based sensing in high-dynamic scenarios.

Table 1. Mean results on synthetic data (incl. blur).

Method	Input	PSNR↑	SSIM↑	LPIPS↓
BlurFrameDGS	BlurRGB	20.32	0.795	0.242
EGrid4D	Event	15.11	0.758	0.266
EGrid4D + new losses	Event	16.77	0.770	0.258
Ours	Event	**22.91**	**0.892**	**0.195**

2.2. Comparison with Existing Baselines

Current dynamic reconstruction frameworks, such as DEGS [10] and Event-boostedDGS [32], typically rely on synchronized RGB-event streams. To provide a meaningful baseline, we adapted Grid4D [31] for event-only supervision (*EGrid4D*). As detailed in Table 1, while the addition of our proposed flow, motion, and depth regularizations improves the performance of the adapted baseline (*EGrid4D + ours*), our full framework significantly outperforms all compared methods. This confirms the effectiveness of our specific architectural choices for single-stream event data.

2.3. Generality of Our Method

The previous point confirms that there is generally merit behind the newly introduced event-based losses and the employed continuous-time formulation, irrespectively of the underlying deformation parametrization (*Ours* or Grid4D). As long as a model supports continuous-time rendering and gradient-based optimization (as is the case with most GS models), our losses are applicable and beneficial (cf. performance improvement of *EGrid4D + new losses* over *EGrid4D* (Table 1).

3. Dataset

3.1. BlenderDynamicEvent

The BlenderDynamicEvent dataset comprises four distinct scenes, whose descriptions are provided below:

Butterfly. The *Butterfly* dataset captures a blue butterfly waving its wings while resting on a flower. The sequence duration is 6.0 seconds.

Duck. The *Duck* dataset features a yellow duck falling towards the ground. The sequence duration is 4.5 seconds.

Alarm. The *Alarm* dataset shows a blue alarm clock jumping and shaking, accompanied by several small characters standing up. The sequence duration is 4.0 seconds.

Ball. The *Ball* dataset consists of eight small balls moving coherently together across a planar surface. The sequence duration is 4.5 seconds.

The monocular camera trajectory corresponding to the four scenes is presented in Figure 1.

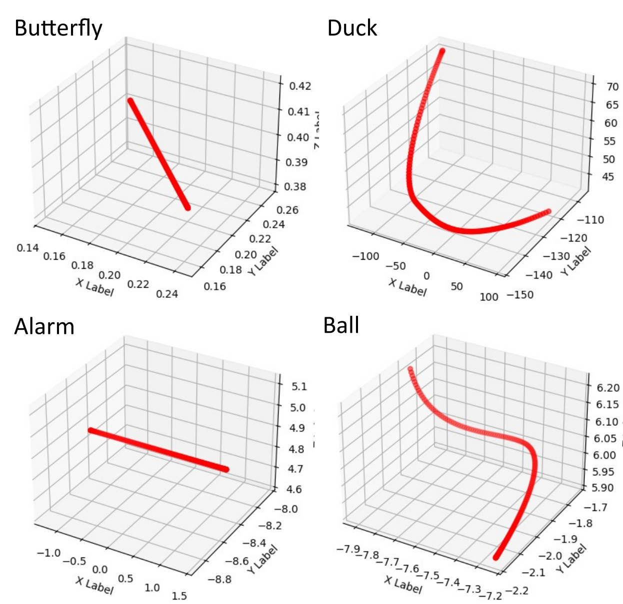

Figure 1. The camera trajectory of BlenderDynamicEvent dataset.

3.2. Gen4Dynamic

The Gen4Dynamic dataset comprises two distinct scenes, whose descriptions are provided below:

Pillow. The *Pillow* dataset captures a pillow falling onto a table. The sequence duration is 1.0 second.

Napkin. The *Napkin* dataset features a napkin falling onto the pillow. The sequence duration is 0.5 seconds.

4. Training Detail

4.1. Hyperparameter

We set the weighting coefficient for the fundamental event loss, λ_e, to 0.2 for all experiments across both datasets. The remaining loss coefficients are tuned based on the characteristics of the data:

- For the synthetic dataset, the hyperparameters were set as follows:$\lambda_{ef} = 1, \lambda_m = 1, \lambda_d = 0.1, \lambda_{ne} = 0.2, \lambda_{sp} = 0.025$
- For the real-world dataset, the hyperparameters were set as follows:$\lambda_{ef} = 1, \lambda_m = 0.1, \lambda_d = 0.1, \lambda_{ne} = 0.1, \lambda_{sp} = 0.025$

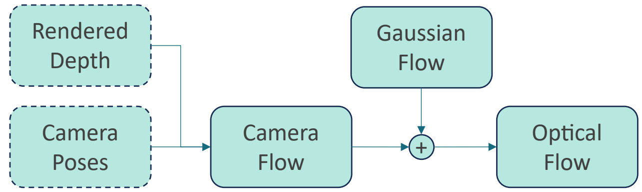

Figure 2. The component of optical flow, which is used to compute flow loss.

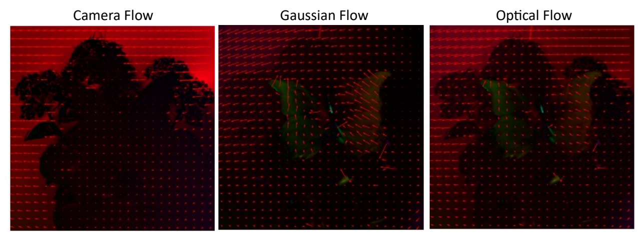

Figure 3. The visualization of camera flow, Gaussian flow, and motion flow on *Butterfly* dataset.

4.2. Gaussian Initialization

Our canonical Gaussians are initialized from a simple cubic point cloud consisting of 10^6 points with randomized parameters.

4.3. Training Cost

Our baseline training time of 0.75h increases to 1.1h with flow and motion losses, and reaches 1.45h with the full depth constraints. The modest 0.7h overhead is a justified trade-off for significant gains in accuracy and consistency, while the B-spline formulation introduces negligible computational cost.

5. Loss Detail

5.1. Flow Loss

The decomposition of the optical flow components used for calculating the flow loss is shown in Figure 2. The visualization of flow components on *Butterfly* is shown in Figure 3.

5.2. Motion Loss

Figure 4 illustrates the impact of the motion loss. Without this constraint, the Gaussians tend to overfit to the training view and exhibit motion that is inconsistent with the event stream.

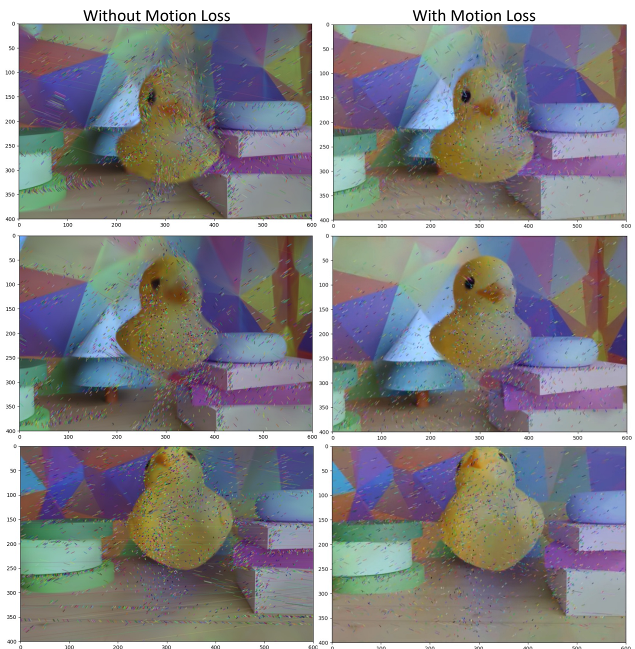

Figure 4. Visualization of Gaussian Trajectories. Comparison of Gaussian trajectories without and with the proposed event motion loss.