Differentiable Event Stream Simulator for Non-Rigid 3D Tracking
—Supplementary Material—

Jalees Nehvi[1,2] Vladislav Golyanik[2] Franziska Mueller[3]
Hans-Peter Seidel[2] Mohamed Elgharib[2] Christian Theobalt[2]

[1]Saarland University [2]MPI for Informatics, SIC [3]Google Inc.

In this document, we provide further details about the proposed event stream simulator including adaptive sampling for parametric models (Sec. A) and application of our approach in tracking fast motions (Sec. B). *See our supplementary video for dynamic visualisations.*

A. Adaptive Sampling

In the case of parametric models like hands, synthetic event generation involves sampling the pose space parameters followed by rendering the corresponding 3D states. The current rendered image and the previously rendered image are subtracted, followed by thresholding. Uniformly sampling the parameters has two main drawbacks. First, it results in redundancy if the sampling rate is too high since there is a small change between the two subsequent images. Secondly, if the sampling rate is too low, a significant portion of relevant event information will be lost.

We apply adaptive policy [2] which tackles this problem by sampling the parameter space intervals according to the time stamps as follows:

$$t_{k+1} = t_k + \lambda\, C \left| max_{\mathbf{x}\in\Omega} \frac{\partial \mathcal{L}(\mathbf{x}; t_k)}{\partial t} \right|^{-1}, \qquad (1)$$

where $\mathcal{L}$ is the difference image intensity, t_k is the current time stamp, $\mathbf{x}$ is the pixel location, t_{k+1} is the next time stamp of the adaptive sampling, Ω is the image plane, $\lambda = 2$ is a constant and C is the contrast sensitivity or threshold and is set to 10. The derivative $\frac{\partial \mathcal{L}}{\partial t}$ is approximated by dividing the difference image by the length of the previous time interval. The new timestamp ensures that very few events are missed out between two consecutively rendered image frames and, at the same time, that any two consecutively rendered frames are non-identical.

B. Application: Tracking Fast Motions

Conventional RGB cameras suffer from motion blur while recording fast motions. Moreover, the use of high-speed cameras for fast motion capture results in a lot of data which need to be stored, compared to an event steam observing the same scene. Since our method operates on events which can capture very fast motions without motion blur and data storage overheads, it can be used in applications that involve tracking fast moving hands.

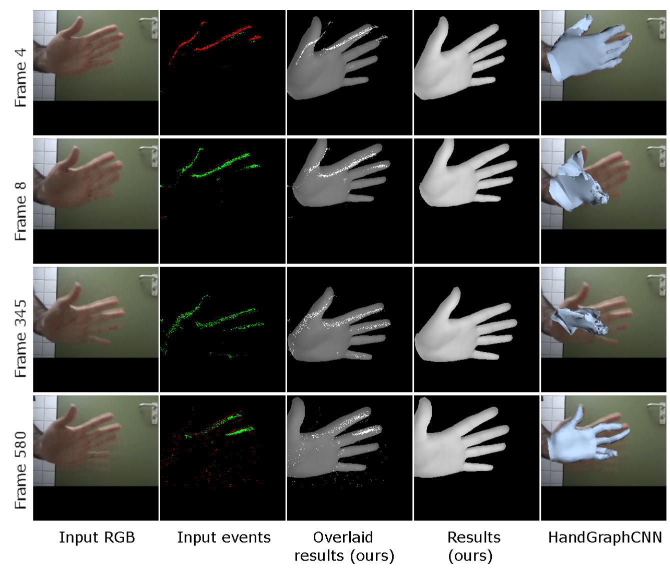

Figure I. Results on the real fast hand sequence. Our method handles fast motion successfully.

We record both RGB images and events of a fast-moving hand using Sony RX0 and DAVIS 240C, respectively. The cameras are time-synchronised using a flash and calibrated (intrinsically and extrinsically). Fig. I compares the qualitative results of our method against HandGraphCNN [1] on fast hand waving motions, see our video for dynamic visualisations. Most RGB images suffer from motion blur and HandGraphCNN [1]—which is an RGB image-based technique—often fails on the blurred images, as one would expect. In contrast, our method produces visually more plausible 3D reconstructions because it operates on events.

References

[1] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[2] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. ESIM: an open event camera simulator. *Conference on Robot Learning (CoRL)*, 2018. 1