Updating Human Pose Estimation using Event-based Camera to Improve Its Accuracy Ippei Otake<sup>1</sup>, Kazuya Kitano<sup>1</sup>, Takahiro Kushida<sup>1</sup>, Hiroyuki Kubo<sup>2</sup>, <sup>1</sup>NAIST <sup>2</sup>Chiba University Akinobu Maejima <sup>3,4</sup>, Yuki Fujimura <sup>1</sup>, Takuya Funatomi <sup>1</sup>, Yasuhiro Mukaigawa <sup>1</sup> <sup>3</sup>OLM Digital, Inc. <sup>4</sup>IMAGICA GROUP, Inc.

## **Objective : Real-time human pose estimation**



- **□** Real-time human interfaces such as live avatar broadcasting and dynamic projection mapping require real-time pose estimation.
- **Current markerless human pose estimation is not** fully real-time.



### References

- 1. Gunnar Farnebäck. 2003. Two-Frame Motion Estimation Based on Polynomial Expansion. In Image Analysis, Josef Bigun and Tomas Gustavsson (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 363–370. 2. Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. 2022. Event-
- Based Vision: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 44, 1 (2022), 154– 180 3. Zicheng Liu. 2019. 3D Skeletal Tracking on Azure Kinect – Azure Kinect Body Tracking SDK. In 3D Computer Vision in Medical Environments in conjunction with CVPR.

### Main issue

**Online pose estimation** degrades accuracy due to pose changes during processing time.

> Improve accuracy by compensating pose changes in processing time.

## Input : Events & Estimated pose Output : Real-time pose

Buffers  $N_b$  events by a fixe number of events.

Extracts only the necessary  $N_1$  events.

### **3.** Dense optical flow calculation :

Estimates the dense optical flow  $F_{m-1:m}$  from two Event images  $I_{mN_b-N_l}$  and  $I_{(m-1)N_b-N_l}$  as an estimate of the joint position changes.

Finally, the  $t_1$ 's pose  $J_0$  is iteratively updated using the dense optical flows, one by one, to

1<sup>st</sup> ~ 3<sup>rd</sup> module of each m<sup>th</sup> phase is processed during runtime of estimation  $t_0 \sim t_1$ , and pose update is processed as soon as finishing estimation at  $t_1$ .



of the operation status from here. Red represents the baseline, blue represents the proposed method, and green represents GT.



SCAN ME

## Experimental results



Blue : pose after updating completion

# The second half has intense motion compare with the first, so the number of updates has increased.





Even in the middle frame with intense movement, our method kept low MPJPE.