Updating Human Pose Estimation using Event-based Camera to Improve Its Accuracy

Ippei Otake
Kazuya Kitano
Takahiro Kushida
otake.ippei.oj2@is.naist.jp
kitano.kazuya@is.naist.jp
kushida.takahiro.kh3@is.naist.jp
NAIST
Japan

Hiroyuki Kubo
hkubo@chiba-u.jp
Chiba University
Japan

Akinobu Maejima
akinobu.maejima@olm.co.jp
OLM Digital, Inc.
IMAGICA GROUP, Inc.
Japan

Yuki Fujimura
Takuya Funatomi
Yasuhiro Mukaigawa
fujimura.yuki@is.naist.jp
funatomi@is.naist.jp
mukaigawa@is.naist.jp
NAIST
Japan

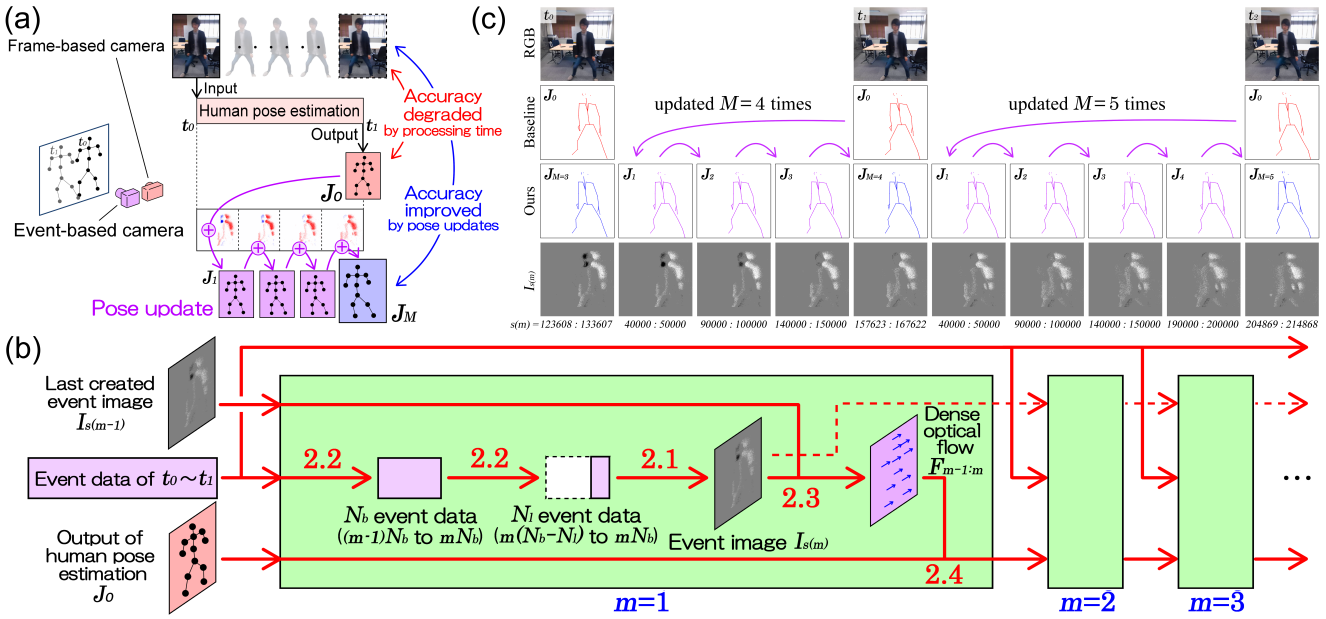

Figure 1: Overview of the proposed method. (a) Pose updates. (b) Proposed architecture. (c) Qualitative evaluation.

ACM Reference Format:
Ippei Otake, Kazuya Kitano, Takahiro Kushida, Hiroyuki Kubo, Akinobu Maejima, Yuki Fujimura, Takuya Funatomi, and Yasuhiro Mukaigawa. 2023. Updating Human Pose Estimation using Event-based Camera to Improve Its Accuracy. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Posters (SIGGRAPH '23 Posters), August 06–10, 2023, Los Angeles, CA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3588028.3603683

1 INTRODUCTION

Real-time human interfaces such as live avatar broadcasting and dynamic projection mapping are becoming increasingly popular. They require real-time pose estimation that can track user performances. Deep learning technologies with a frame-based camera make this possible, but the latency caused by the processing time in pose estimation degrades accuracy. In particular, when the user performs quickly, the estimated pose is already different at that time, leading to degradation in the accuracy of the pose estimation.

In this paper, we propose a method to improve the accuracy degraded by latency of the pose estimation by using an event-based camera. A event-based camera outputs changes in luminance values with low latency and high temporal resolution [Gallego et al. 2022].

2 OUR APPROACH

Let t_0 be the time when a person image is captured by a frame-based camera, and pose estimation is applied to this image. However, the pose estimation is completed at t_1 due to the processing time. Our objective is to estimate the pose at t_1 by updating the estimate from the pose estimator with event data acquired during $[t_0, t_1]$. Figure 1(a) shows the updating procedure proposed in this paper.

SIGGRAPH '23 Posters, August 06–10, 2023, Los Angeles, CA, USA

ACM ISBN 979-8-4007-0152-8/23/08.
https://doi.org/10.1145/3588028.3603683

The proposed method takes event data as information about pose changes along time and updates the joint positions accordingly. The proposed method updates the joint positions gradually when a sufficient amount of the event data has been obtained, rather than accumulating the data during $[t_0, t_1]$ and processing it all at once.

2.1 Event stacking

This process converts the event data into an event image. Event data $\boldsymbol{E}_{n:m} = \{E_i \mid n \leq i < m\}$ is output from the event-based camera in the format $E_i = (x, y, p, t)$. Here n, m are the IDs of the event data, t is the time, and p is a binary output indicating whether the intensity of the pixel at (x, y) has changed positively or negatively. The event image $I_{n:m}$ is an accumulation of the $\boldsymbol{E}_{n:m}$ as a single image with three different values as follows:

$$I_{n:m}\left[x_0, y_0\right] = \begin{cases} 255 & (x_0, y_0, \mathrm{Pos}, t) \in \boldsymbol{E}_{n:m} \\ 127 & (x_0, y_0, p, t) \notin \boldsymbol{E}_{n:m} \\ 0 & (x_0, y_0, \mathrm{Neg}, t) \in \boldsymbol{E}_{n:m} \end{cases}. \tag{1}$$

2.2 Adaptive event buffering

In the case of converting event data into an event image per a fixed time interval, the event image becomes redundant with less information when the number of events during the interval is short. Conversely, when the number of events is large, the event image continuously forms the motion trajectories. This corresponds to motion blur in a frame-based camera, which leads to low accuracy in the subsequent estimation process. Therefore, we design this process to be adaptive to the motion by buffering every N_b events into an event image. In addition, we propose to use only the latest N_l event data from the buffered N_b event data to make the estimation accurate. This buffering results in the event image forming the contour of the body parts where the motion occurs. We also expect this to reduce the computational cost. Eventually, The event image $I_{s(m)}$ is generated, where $s(m) = mN_b - N_l : mN_b$.

2.3 Dense optical flow calculation

This process estimates the dense optical flow, which is a two-dimensional velocity field per pixel, from two successive images as an estimate of the joint position updates. This paper uses the Farneback method [Farnebäck 2003], which is a simple but accurate, to generate the dense optical flow $F_{m:m+1}$ from $I_{s(m)}$ and $I_{s(m+1)}$.

2.4 Pose update

At t_1, when the pose estimation is complete, the joint position that should have been at t_0 is estimated. Let $\boldsymbol{J}_0 = \left\{J_0^k = (x_k, y_k) \mid k \in K\right\}$ be the joint position at t_0, which is the pose estimation result $\boldsymbol{J}_0$, where k is the joint ID. Suppose that M dense optical flows are obtained during $[t_0, t_1]$. $\boldsymbol{J}_0$ is iteratively updated using the dense optical flows, one by one, to estimate the joint positions $\boldsymbol{J}_M$ at t_1 as follows:

$$J_{m+1}^k = J_m^k + F_{m:m+1}\left[x_k, y_k\right] (m = 0, \cdots, M-1) \tag{2}$$

3 EXPERIMENTS

For evaluation, a performance was recorded as a dataset and the proposed method was applied offline. In the performance, a human moved left and right. We used a Prophesee Gen3 event-based camera and an Azure Kinect as a frame-based camera to capture the performance. The Azure Kinect Body Tracking [Liu 2019] is used for a frame-based pose estimation. Event buffering was performed every $N_b = 50,000$ and selected to $N_l = 10,000$. Since the pose estimation is performed offline, the ground truth of the joint positions is obtained from the frame-based images. In the evaluation, we also use the computation time of the pose estimation to simulate the latency and use it as a baseline. We applied the proposed method to update the pose estimation output using the event data.

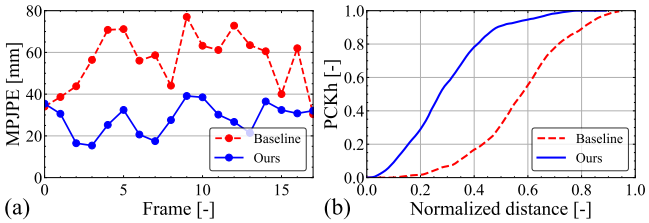

Figure 2: Quantitative evaluation. The MPJPE transition in consecutive frames is shown in (a) and PCKh is shown in (b).

Figure 2(a) shows a quantitative evaluation using MPJPE, which represents the mean error of the joints. In the baseline, the MPJPE increased as the motion became more intense. Meanwhile, the proposed method could keep the MPJPE below a certain level. The accuracy was comparable around frames 0 and 17, when the motion was small, but significantly improved for the intense motion.

Figure 2(b) shows the quantitative evaluation using PCKh, which expresses the percentage of the correct keypoints as whether the estimated position is within the radius expressed as the ratio of the joint length from the neck to the head. The proposed method shifts the plot significantly to the left compared to the baseline output. The percentage of the correct estimates improves by a factor of 2 to PCKh@0.5. As well as the MPJPE, the proposed method was effective in keeping the PCKh below a certain level.

The results of the proposed method are shown in Fig. 1(c) with the bones colored in purple. The second half has intense motion, and the number of updates has increased compared to the first half.

4 CONCLUSION

Experimental results showed that the proposed method improves the accuracy by pose updating during the pose estimation process. A future challenge includes the verification in online processing.

ACKNOWLEDGMENTS

This work was partly supported by JST PRESTO JPMJPR2025 and JSPS KAKENHI Grant Number JP23K16902.

REFERENCES

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.

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.

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.*