Event-based Robust 3D Pose Estimation Using Time Series Data

Kakeru Koizumi
Graduate School of Fundamental Science and Engineering, Waseda University
Tokyo, Japan
kkeverio@ruri.waseda.jp

Hiroshi Watanabe
Graduate School of Fundamental Science and Engineering, Waseda University
Tokyo, Japan
hiroshi.watanabe@waseda.jp

2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC) | 979-8-3503-8459-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/AIC61668.2024.10731089

***Abstract*—Event cameras are vision sensors that detect asynchronous changes in luminance for each pixel. They are effective for 3D pose estimation in poorly illuminated environments since they have a wider dynamic range than conventional RGB cameras. Hence, they are expected to be used as surveillance cameras for detecting suspicious persons, especially at night. However, practical applications are hindered by the high cost of event cameras and the difficulty of their synchronization. To address the limited widespread use of event cameras, we ensure practicality by implementing monocular pose estimation. Common methods for event-based pose estimation involve creating a frame that combines a set number of asynchronous events. With these methods, only changes in motion can be captured due to the nature of event cameras. This makes estimation unstable since information on joints that do not move is not collected. Therefore, we propose a stable pose estimation method that accumulates event information by incorporating past time series data. In addition, datasets for event-based pose estimation, especially those consisting of raw event data, are rare and lack diversity. To overcome the lack of data, we use data augmentation to create a robust event dataset for person localization and size estimation. Incorporating past time series data along with data augmentation enhances the versatility and accuracy of event-based monocular pose estimation.**

***Index Terms*—3D Human Pose Estimation, Event-based Vision, Data Augmentation, Temporal Convolutions**

I. Introduction

Event cameras [1], [2] are a dynamic vision sensor (DVS) inspired by the retina of living organisms. Unlike conventional RGB cameras, it detects changes in the luminance of each pixel, outputting ±1 if the change exceeds a certain threshold and 0 if it does not. [The output of an event camera is generally referred to as an event.] Event cameras have a wider dynamic range than conventional cameras, allowing them to accurately capture the outlines of subjects at night. Hence, event cameras effectively estimate a person's posture even in poorly illuminated environments. This capability makes event cameras highly promising for the nighttime detection of suspicious individuals and for maintaining public safety as surveillance cameras. However, event cameras face several obstacles in practical implementation. Their widespread adoption is hindered by the high cost and the difficulty of precisely synchronizing multiple units. Given these circumstances, we aim to enhance the versatility of monocular pose estimation and promote its widespread use.

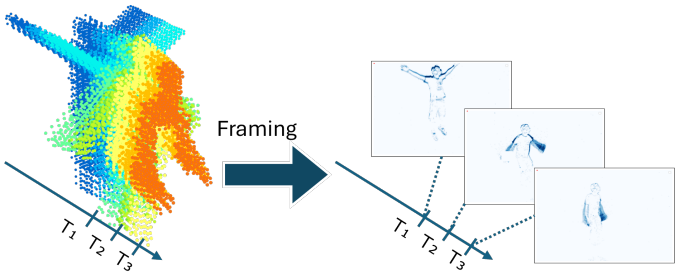


Fig. 1: Framing of event data acquired from event cameras.

Common approaches to event-based pose estimation involve the Frame-based method [3], [4]. This method divides events at fixed intervals, accumulates each event, and transforms them into an image, as shown in Figure 1. However, since the event camera captures only the differences in motion, some joints are unable to acquire enough events in scenes with unbalanced motion, such as the T_3 posture in Figure 1. This leads to a decrease in the accuracy of whole-body pose estimation. In contrast, some methods address this issue by incorporating past and future information [5]. Joints that could not be acquired in a single frame can be regarded as motionless, so posture information detected in the frames immediately before and after is used to supplement the whole-body posture and stabilize accuracy.

In this paper, we propose a pose estimation method using current and past information. This study does not introduce the use of future information as it hinders real-time pose estimation. By accumulating events from the past few frames, we can supplement joint positions that cannot be detected in the current single frame. We aim to improve the accuracy of pose estimation by incorporating past information using the Long Short-Term Memory (LSTM) module.

In addition to introducing LSTM into the pose estimation model, we also address data augmentation. DHP19 dataset is the first dataset consisting solely of event data with 3D pose information and has been widely used in studies on event-based pose estimation. However, DHP19 contains many frames where a simply moving subject is centered in the field of view, which is quite different from the frames in

979-8-3503-8459-8/24/$31.00 ©2024 IEEE
DOI: 10.1109/AIC.2024.200

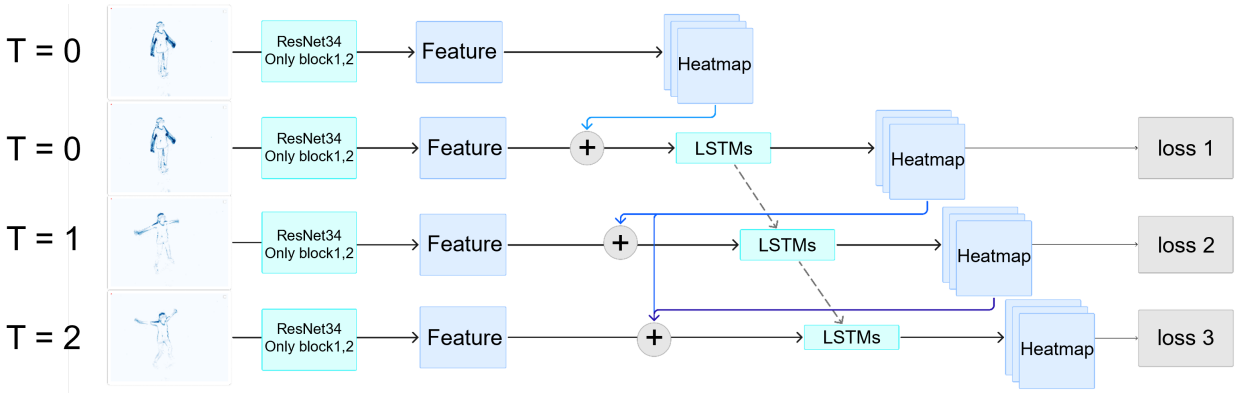


Fig. 2: The overall network structure of our proposed method.

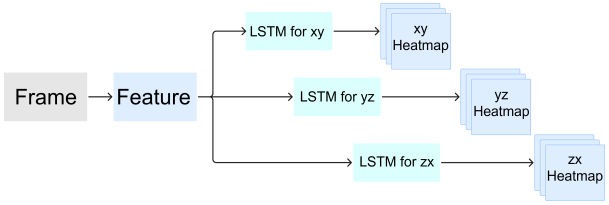


Fig. 3: The detailed structure for generating heat maps.

actual use cases, such as surveillance videos. In fact, the pose estimation model trained on this dataset is significantly less accurate for frames where the subject is at the edge of the angle of view. To address this issue, we propose a method to augmented DHP19 to be more robust to such scenes by creating data with randomly processed subject positions and sizes. This data augmentation allows for training models that are robust to changes in the position and size of the person. Finally, by introducing past time series data into the augmented dataset, we develop a model that is robust to various scenes and addresses the shortcomings of frame-based estimation methods.

In summary, our proposal consists of two major contributions:

- Improving accuracy by incorporating past information and connecting time-series data.
- Creating an augmented dataset that maintains stable pose estimation accuracy regardless of the subject's location and size, thereby improving the versatility of event-based monocular 3D pose estimation.

II. Related Works

A. Frame-based Pose Estimation

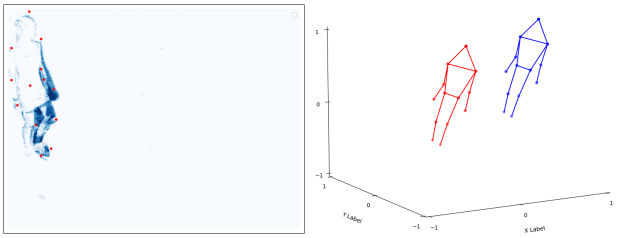
Event cameras output a sequence of event signals asynchronously. Mainly, the event stream is segmented and framed to treat like an image in event-based pose estimation. The frame-based method can be adapted for conventional image processing tasks and is capable of handling various models [3], [7]. As a splitting method for the event stream, we employ a method that divides it into a constant number of events, as in [7]. In our study, 7500 events are aggregated in each frame to construct a synchronous event representation. However, in this method, scenarios where the subject exhibits biased movements, such as moving only the arms, make it difficult to detect non-moving joints. In addition, frames may occur in which events from those joints are not sufficiently accumulated. Consequently, there arises an issue of decreased accuracy in whole-body pose estimation.

B. DHP19

DHP19 [6] is the first dataset for 3D pose estimation consisting only of event data. 33 scenes of 17 subjects are recorded. Each scene consists of 10 repetitions of a particular action. All subjects are recorded using four event cameras positioned at different orientations. The field of event-based pose estimation is still in its early stages, and there are only a few publicly available datasets. Thus, some datasets [8], [12] have converted RGB images to look like event data. DHP19 which records raw event data and includes 3D posture information is highly valuable. However, as pointed out in a conventional study [9], most of the movements recorded in DHP19 are simple and lack significant motion. These characteristics limit its versatility for real-world applications.

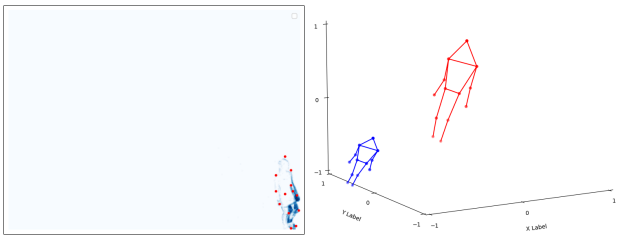
C. Margi pose Model

Margi pose model estimates the z-axis from a single frame and generates Marginal Heatmaps [10], [11] in the xy, yz, and xz planes. Only the first and second blocks of ResNet [13] are used to extract features for heatmap creation. Creating xy, yz, and zx heatmaps from the extracted features is considered one stage. To improve accuracy, the features and the output heatmaps are combined, and this stage is repeated three times. When converting event streams into frames, the 3D posture is transformed into the viewpoint coordinate system P_{xyz}, which is then projected onto the image plane using a perspective projection transformation. The depth z_{ref} of the image plane is defined by the P_z coordinate of the subject's head. The projected image plane is then converted to the normalized cube P_{xyz}^{NDC}(Normalized Device Coordinates [14], [15]) to obtain 3D normalization information.



(a) An input frame with subject translated. (b) Normalized posture information.

Fig. 4: The input frame and normalized 3D posture information with a parallel shift applied to the subject. (Red: before translation, Blue: after translation)



(a) An input frame with subject resized. (b) Normalized posture information.

Fig. 5: The input frame and normalized 3D posture information with a reduction process applied to the subject. (Red: before translation, Blue: after translation)

D. Feature Concatenated Model

Feature concatenated model incorporates time series information, addressing the issues associated with frame-based pose estimation described in the previous section [5]. By alternately concatenating features from adjacent frames in the horizontal direction, the model achieves both improved accuracy and reduced computational complexity. However, this model faces difficulty in adapting time series information to vertically moving postures. Additionally, it suffers from low real-time performance due to the incorporation of future information.

E. 2D event-based pose estimation model with LSTM modules

This model [9] improves the accuracy of event-based 2D pose estimation by using an LSTM module. The LSTM employs two convolutional LSTM layers [16] without peephole connections to store event information from a series of five frames.

III. Proposed Model

A. LSTM Based Neural Network

We propose an event-based monocular 3D pose estimation model that introduces LSTM as the connection of time series data. The details are shown in Figure 2. The LSTM module is inserted during the process of creating a heatmap from the features of each frame. The heatmap up to the previous frame and the feature of the current frame are used as input for the LSTM. Since there is no previous heatmap to be input to the LSTM module for the first frame, an initial heatmap is created without using the LSTM. Then, this heatmap and the features of the first frame are used as input for the initial LSTM module. In addition, as shown in Figure 3, an LSTM was employed in the creation of each XY, YZ, and ZX heatmap to enhance the accuracy of depth estimation in the direction. The loss function is given in the following equation.

$$L = \sum_{t} L_{geometrical}\left(p^{t}_{xyz}, \hat{p}^{t}_{xyz}\right) + JSD\left(H^{t}_{xy}, \hat{H}^{t}_{xy}\right) + JSD\left(H_{yz}, \hat{H}^{t}_{yz}\right) + JSD\left(H_{zx}, \hat{H}^{t}_{zx}\right) \quad (1)$$

For each successive frame, the loss is calculated as the sum of the squared errors between the ground truth 3D camera coordinates p^{t}_{xyz} and the predicted 3D coordinates $\hat{p}^{t}_{xyz}$, along with the sum of the Jensen-Shannon divergence $JSD(H, \hat{H})$ for each heatmap $(H_{xy}H_{yz}, H_{zx})$. Here, H is the ground truth heatmap, and $\hat{H}$ is the predicted heatmap.

TABLE I: Pose estimation results for each model on the augmentation dataset and DHP19

Method	Stage	Train	Test	
			Aug Dataset	DHP19
Scarpellini *et al.* [12]	1	DHP19	383.72	93.82
Scarpellini *et al.* [12]	3	DHP19	265.10	91.56
Proposed method (LSTM)	1	DHP19	232.78	90.15
Proposed method (Aug)	1	Aug Dataset	106.97	90.88
Proposed method (Aug)	3	Aug Dataset	88.71	86.19
Proposed method (Aug+LSTM)	1	Aug Dataset	**87.03**	**84.69**

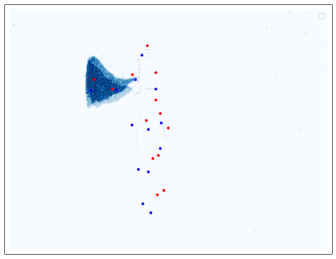
B. Data Augumentation

We propose a data augmentation method that involves translating and scaling the event data of input frames to enhance the versatility of the dataset. When processing the dataset, as illustrated in Figure 4(a), we randomly shift the subjects in the input frames horizontally and vertically, ensuring that all their joints remain within the frame. Additionally, as shown in Figure 5(a), we randomly apply scaling with a certain probability to accommodate subjects of various sizes. Along with processing the input frames, as indicated in Figures 4(b), 5(b), we normalize the amount of pixel movement of the joints and apply the same translation and scaling to the posture information within the normalized cube, P^{NDC}_{xyz} [14], [15].

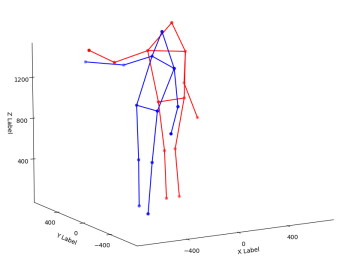
As mentioned in the previous section (section II.C), the normalization of information using the joint projection matrix is based on the depth coordinate z_{ref} of the subject's head. By fixing z_{ref} during the translation and scaling transformations, we ensure that no depth displacement occurs due to processing.

IV. Experiments

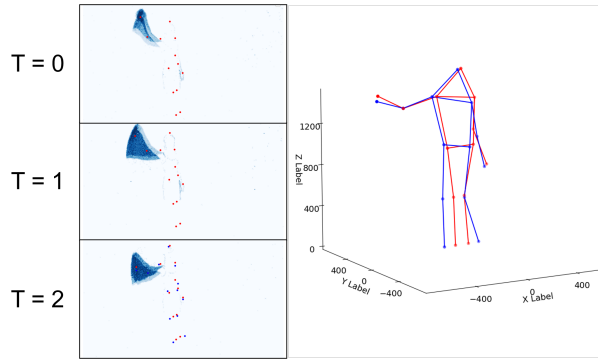
We experiment with three patterns of the method proposed in the previous section. First, we compare the model using the



(a) Conventional 2D pose estimation result.



(b) Conventional 3D pose estimation result.



(c) The pose estimation results using the LSTM module. (Left: input frames, Right: 3D pose estimation results)

Fig. 6: Comparison of pose estimation results between conventional methods and our method incorporating past information

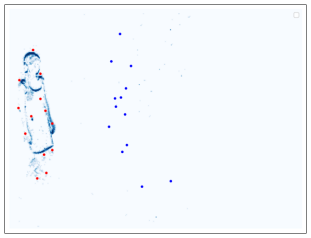
LSTM module with the conventional method to confirm the validity of the time series data. Second, we demonstrate the improvement in versatility by using a dataset with subjects extended to a variety of sizes and locations. Finally, we conduct experiments using both methods simultaneously to show the improvement in accuracy and generalizability. In every experiment, we used Mean Per-Joint Position Error (MPJPE) as the evaluation metric.

A. LSTM Based Neural Network

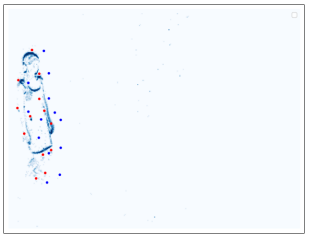
The proposed method incorporating the LSTM module is compared with the conventional method, with the quantitative results presented in Table I. Compared to the conventional method in terms of pose estimation accuracy averaged over the entire dataset, our method shows only a slight improvement. However, for specific movements, such as ”Left leg abduction” and ”Right leg knee lift” in Table II, there is a noticeable improvement in accuracy. As shown in Figures 6(a), 6(b), the conventional method cannot estimate the full-body 3D posture for frames where only the right arm is detected. In contrast, our method improves the accuracy of whole-body pose estimation by incorporating past frames and accumulating events from the left arm and left hip, as shown in Figure 6 (c).

B. Data Augumentaion

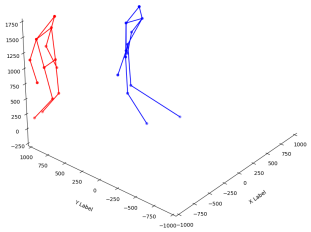
The table I shows the results of training and testing on the DHP19 dataset and the augmented dataset, respectively, using the model from the previous study. The extent of subject



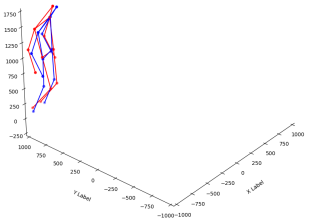
(a) Conventional method.



(b) Proposed method.

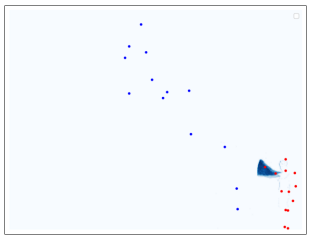


(c) Conventional method.



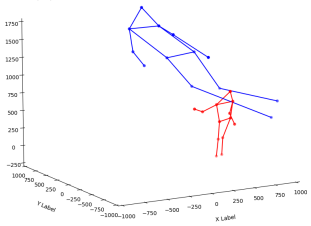
(d) Proposed method.

Fig. 7: Comparison of 2D and 3D pose estimation results for frames where the person is moved to the left (Red: ground truth, Blue: estimated)

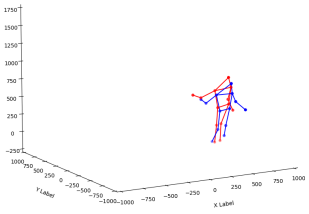


(a) Conventional method.

(b) Proposed method.



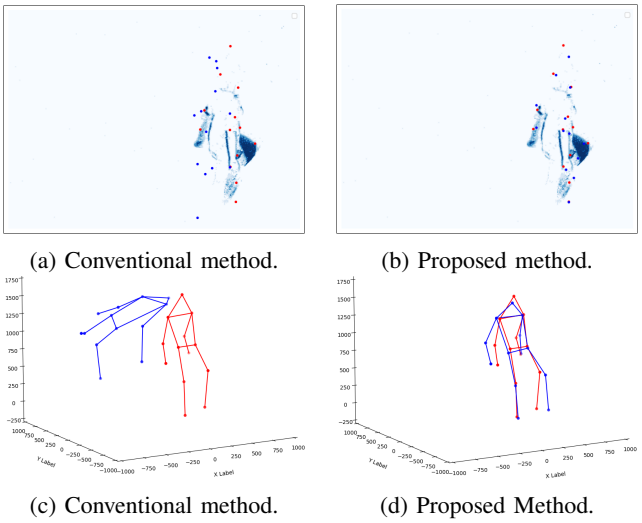
(c) Conventional method.



(d) Proposed method.

Fig. 8: Comparison of 2D and 3D pose estimation results for frames with reduced subject size. (Red: ground truth Blue: estimated)

movement in each frame is random, following a uniform probability distribution. Additionally, 20% of the total data underwent a reduction process before being shifted in parallel. As shown in Figures 7, 8, the conventional study's model is unable to estimate posture for frames where the subject is at the edge of the input image or when the subject is small. In contrast, the model trained on our augmented dataset is able to estimate posture with sufficient accuracy. The model trained on the augmented dataset also achieves higher accuracy than the model from the conventional study for DHP19. Particularly, this method is effective in scenes where the subject is at the edge of the frame, such as in “Multiple Jumps Up Down” and



(a) Conventional method. (b) Proposed method.

(c) Conventional method. (d) Proposed Method.

Fig. 9: Comparison of pose estimation results for DHP19 between the conventional method and the model trained on the our augmented dataset (Red: ground truth, Blue: estimated)

TABLE II: Comparison of the accuracy of each model in scenes recorded in DHP19

Scene	Conv [12]	LSTM	Aug	Aug+LSTM
Left arm abduction	73.93	**73.09**	74.20	73.65
Right arm abduction	79.35	79.17	79.16	**76.59**
Left leg abduction	108.41	104.22	105.72	**102.81**
Right leg abduction	96.00	95.08	92.69	**90.58**
Left arm bicep curl	77.39	**75.15**	76.35	75.51
Right arm bicep curl	85.01	**81.11**	86.33	84.42
Left leg knee lift	82.08	82.51	79.39	**76.97**
Right leg knee lift	79.71	**74.62**	79.57	77.87
Walking 3.5km/h	87.61	86.27	78.14	**75.80**
Single jump up-down	76.34	77.45	72.30	**71.26**
Single jump forwards	83.70	84.55	79.38	**78.23**
Multiple jumps up-down	106.31	102.44	82.65	**82.16**
Hop right foot	93.34	93.81	90.58	**89.55**
Hop left foot	98.49	94.76	91.30	**89.92**
Punch straight forward left	92.31	89.18	88.48	**86.32**
Punch straight forward right	80.91	83.27	80.20	**78.70**
Punch up forwards left	97.77	98.01	94.41	**92.54**
Punch up forwards right	92.25	90.09	86.85	**86.30**
Punch down forwards left	86.23	**85.08**	87.23	85.84
Punch down forwards right	78.99	**77.42**	82.18	80.68
Slow jogging 7km/h	86.32	85.07	84.33	**84.29**
Star jumps	107.31	105.42	94.93	**93.11**
Kick forwards left	99.88	96.64	93.79	**91.91**
Kick Forwards right	106.52	102.2	102.44	**101.00**
Slide kick forwards left	127.68	127.65	124.05	**122.65**
Slide kick forwards right	115.59	115.30	109.64	**107.90**
Wave hello left hand	87.59	**84.50**	89.40	87.66
Wave hello right hand	71.05	**70.69**	74.59	73.80
Circle left hand	82.24	80.40	79.50	**79.47**
Circle right hand	79.79	78.99	79.71	**78.80**
Figure-8 left hand	81.01	79.62	81.33	**79.56**
Figure-8 right hand	76.11	73.9	75.79	**73.81**
Clap	78.72	74.63	75.07	**74.58**
average	91.56	90.15	86.19	**84.91**

"Star Jumps" in Table II, demonstrating higher accuracy than the conventional method.Qualitatively, as shown in Figure 9, our model can accurately estimate posture for frames recorded in DHP19.

C. Combined Method

Finally, we present the estimation results for the model combining the two methods mentioned above. As shown in Table I, our method, trained on the augmented dataset and incorporating LSTM modules, achieves better accuracy than the conventional method. Additionally, this model achieves the best accuracy compared to models adapted from each of the two proposed methods on both the Aug Dataset and DHP19.

V. Conclusion

In this paper, we propose an event-based monocular 3D pose estimation model. Our model incorporates past information and is robust to variations in subject position and size. Compared to conventional methods, our approach improves the accuracy of pose estimation, particularly for subjects with minimal motion and across a wide range of positions and sizes. However, our model does not fully exploit the features of event data in its use of time-series information. The current model outputs all joints regardless of the collected events, making it unable to identify joints for which no events have been detected. To address this, we are considering enhancing the model by introducing a confidence level, similar to those used in RGB-based pose estimation.

References

[1] G. Gallego *et al.*, "Event-based vision: A survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 154-180, Jul. 2020.

[2] L. Patrick, C. Posch, and T. Delbruck, "A 128x 128 120 db 15μs latency asynchronous temporal contrast vision sensor," IEEE Journal of Solid-State Circuits, vol. 43, pp. 566–576, 2008.

[3] A. Tabia, F. Bonnardi, and S. bouchafa, "Deep Learning For Pose Estimation From Event," International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp. 1-7, Feb. 2023.

[4] C. Boretti *et al.*," PEDRo: an Event-based Dataset for Person Detection in Robotics," IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 4065-4070, Jun.2023.

[5] K. Koizumi and H. Watanabe, "3D Pose Estimation Using Time Series Data in Event-based Video," The 8th IIEEJ International Conference on Image Electronics and Visual Computing (IEVC 2024), Mar. 2024

[6] E. Calabrese *et al.*, "Dhp19: Dynamic vision sensor 3d human pose dataset," IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1695-1704, Jun. 2019

[7] D. P. Moeys *et al.*, "Steering a predator robot using a mixed frame/event-driven convolutional neural network," In International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP), pp. 1-8, Oct. 2016.

[8] G. Goyal *et al.*, "MoveEnet: Online High-Frequency Human Pose Estimation with an Event Camera," IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 4024-4033, Aug. 2023.

[9] Z. Shao *et al.*, "A Temporal Densely Connected Recurrent Network for Event-based Human Pose Estimation," Pattern Recognition, vol 147, Mar. 2024.

[10] D. Mehta *et al.*, "Monocular 3d human pose estimation in the wild using improved cnn supervision," 2017 International Conference on 3D Vision (3DV), pp. 506-516, Oct. 2017.

[11] A. Nibali *et al.*, "3D Human Pose Estimation with 2D Marginal Heatmaps," Jun. 2018.

[12] G. Scarpellini, P. Morerio, and A. D. Bue, "Lifting monocular events to 3d human poses," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1358-1368, Jun. 2021

[13] K. He *et al.*, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, Jun. 2016

[14] T. McREYNOLDS and D. BLYTHE," Advanced Graphics Programming Using OpenGL, The Morgan Kaufmann Series in Computer Graphics," pp. 19–34, 2005.

[15] S. H. Ahn. OpenGL projection onto frustum space.

[16] X. Shi *et al.*, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," Advances in Neural Information Processing Systems 28 (NIPS), 2015.