EventPointMesh: Human Mesh Recovery Solely From Event Point Clouds

Ryosuke Hori, *Student Member, IEEE*, Mariko Isogawa, *Member, IEEE*, Dan Mikami, *Member, IEEE*, and Hideo Saito, *Senior Member, IEEE*

***Abstract*—How much can we infer about human shape using an event camera that only detects the pixel position where the luminance changed and its timestamp? This neuromorphic vision technology captures changes in pixel values at ultra-high speeds, regardless of the variations in environmental lighting brightness. Existing methods for human mesh recovery (HMR) from event data need to utilize intensity images captured with a generic frame-based camera, rendering them vulnerable to low-light conditions, energy/memory constraints, and privacy issues. In contrast, we explore the potential of solely utilizing event data to alleviate these issues and ascertain whether it offers adequate cues for HMR, as illustrated in Fig. 1. This is a quite challenging task due to the substantially limited information ensuing from the absence of intensity images. To this end, we propose EventPointMesh, a framework which treats event data as a three-dimensional (3D) spatio-temporal point cloud for reconstructing the human mesh. By employing a coarse-to-fine pose feature extraction strategy, we extract both global features and local features. The local features are derived by processing the spatio-temporally dispersed event points into groups associated with individual body segments. This combination of global and local features allows the framework to achieve a more accurate HMR, capturing subtle differences in human movements. Experiments demonstrate that our method with only sparse event data outperforms baseline methods.**

***Index Terms*—Event camera, human mesh recovery, human pose and shape estimation, point cloud.**

I. Introduction

With the rise of virtual reality (VR), augmented reality (AR), metaverse applications, and other immersive experiences like sports viewing and stage performances, understanding a human pose and shape non-invasively has become paramount. These immersive applications not only enhance our digital social interactions but also revolutionize content creation, animation, training, and rehabilitation. However, the challenge is to capture human states, including 3D pose and shape, accurately across various real-world settings to elevate the depth of the immersion in these digital realms.

So far, optical-based motion capture (MoCap) systems [1], [2] have offered high-precision and high-speed capturing capabilities. However, their primary challenge remained the associated high costs. With the progression of technology, there has been a notable drift towards data-driven pose estimation methods [3], [4], [5], with techniques leveraging affordable RGB cameras gaining traction. These pose estimation techniques predominantly work with sparse 3D joint representations in the form of skeletal models. While these skeletal models can represent simple actions relatively clearly, they fall short when detailing intricate human behaviors. The need to describe the human body with finer granularity has led to a renewed focus on HMR methods [6], [7], [8]. HMR aims to delve deeper, estimating intricate details of human form, including the 3D pose and shape. However, a significant challenge with RGB-based methods remains their heavy dependence on visible light, which leads to issues such as object occlusions, lighting condition constraints, privacy issues in some scenarios, and considerable power and memory demands hindering edge device deployment. As a solution, researchers have been exploring HMR techniques using alternative modalities such as wireless signals. Notably, methods leveraging radio frequency (RF) signals, including Wi-Fi and millimeter-wave (mmWave) [9], [10], have gained attention. These methods take advantage of longer wavelengths than visible light, reducing memory and personal data consumption. However, they pose challenges, including potential restrictions in environments with sensitive electronic equipment, such as hospital rooms or aircraft.

The use of event-based cameras, henceforth event cameras, has the potential to be a solution to these challenges. Unlike conventional cameras, event cameras detect changes in scene brightness, generating event data including coordinates, time, and polarity. They independently monitor luminance changes for each pixel, recording only significant changes asynchronously. This mechanism ensures high temporal resolution and low power consumption. Furthermore, focusing solely on brightness changes, event cameras have a high dynamic range, making them resilient in low-light conditions, as shown in Fig. 1(a). From these characteristics, event cameras have been incorporated

Received 6 November 2023; revised 18 August 2024; accepted 12 September 2024. Date of publication 18 September 2024; date of current version 1 August 2025. This work was supported in part by JST PRESTO under Grant JPMJPR22C1, in part by the Keio University Research Promotion Fund, JSPS Research Fellowships for Young Scientists DC1, and in part by JSPS Grant-in-Aid for Scientific Research (B) under Grant 23H03422. Recommended for acceptance by Y.-Y. Chuang. *(Corresponding author: Ryosuke Hori.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Bioethics Committee of the Graduate School of Science and Technology at Keio University under Application No. 2023-112.

Ryosuke Hori, Mariko Isogawa, and Hideo Saito are with the Department of Information and Computer Science, Keio University, Yokohama, Kanagawa 223-8522, Japan (e-mail: hori-rysk@keio.jp).

Dan Mikami is with the Department of Computer Science, Kogakuin University, Shinjuku, Tokyo 163-8677, Japan.

The dataset and code will be available at https://github.com/RyosukeHori/EventPointMesh.

This article has supplementary downloadable material available at https://doi.org/10.1109/TVCG.2024.3462816, provided by the authors.

Digital Object Identifier 10.1109/TVCG.2024.3462816

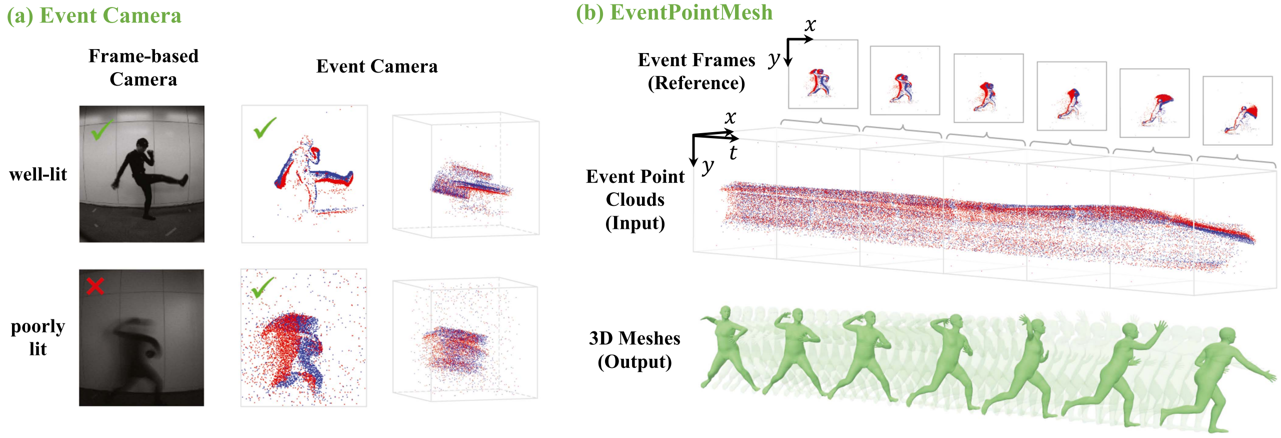

Fig. 1. We propose EventPointMesh, a method for 3D human mesh recovery solely from event data. (a) Event cameras are designed to detect luminance changes, offering high temporal resolution and a high dynamic range. These unique features effectively address the challenges faced by frame-based cameras, such as motion blur and low frame rates in poorly-lit environments. (b) EventPointMesh treats event data as point clouds and estimates human meshes from these point clouds segmented by fixed-time intervals. It enables high-speed HMR that is unaffected by lighting conditions.

into various research areas, including tracking, recognition, 3D reconstruction, robotics, VR/AR, and autonomous driving [11]. Our focus is on employing event cameras for HMR. While there have been proposals for HMR using event data [12], [13], these methods utilize grayscale frame images captured simultaneously with event data for initial pose and shape estimation at each time step. This limits the anticipated resilience to dark conditions and energy/memory efficiency of event-based methods.

In this paper, we tackle the challenge of HMR using only event data, aiming to develop an HMR method that can operate under various lighting conditions, consumes less power and memory, and also preserves privacy. To this end, we propose EventPointMesh, a framework designed to estimate human meshes from event data interpreted as 3D spatiotemporal point clouds. As illustrated in Fig. 1(b), the event data consists of points distributed over the 3D spatiotemporal domain, spanning the xy-axes of the image and the temporal t-axis, henceforth referred to as the “event point cloud”. Each point possesses a polarity information indicating the direction of luminance change; the red points in the figure represent pixels where the luminance has increased, while blue denotes pixels where it has decreased. In EventPointMesh, streams of event point clouds are segmented into fixed time windows and fed into the network as blocks of point clouds for processing. Within the network, a two-stage feature extraction process is conducted on these point cloud blocks to estimate human meshes, traversing through multiple modules. The first stage involves extracting coarse global features from the point clouds, which contain information about the human’s location, body shape, and approximate pose. The next stage involves a mechanism for extracting fine point cloud features to precisely replicate the human pose. Initially, a module within the network estimates 2D joint positions on the image plane. Then, for each joint individually, another module groups the event points around that specific joint and extracts local features for each of these groups. By combining the global and local features obtained separately, the method ultimately estimates the human’s accurate pose, body shape, and location in 3D space. This coarse-to-fine feature extraction approach enables the retrieval of useful information solely from point clouds, allowing this method to faithfully depict human dynamics from sparse event point clouds without the need for intensity images. Additionally, the event point clouds input to the EventPointMesh network can be set at any temporal width, harmonizing with the high temporal resolution characteristic of event cameras. This facilitates the realization of the HMR method capable of restoring high-frequency 3D poses and shapes of diverse movements in poor lighting conditions.

Furthermore, we propose EventPointMesh Dataset (EPMD), a large-scale HMR dataset comprising event data, intensity images, optical MoCap data, and human mesh models. While there have been event-based HMR datasets before, what sets EPMD apart is its capturing of diverse motion and shape data in both well-lit and poorly-lit conditions. This is the first dataset of its kind to incorporate such lighting variations. Extensive experimental evaluations using the existing large-scale event-based HMR dataset, Multi-Modality Human Pose and Shape Dataset (MMHPSD) [13], and our EPMD, under both well-lit and poorly-lit conditions, have validated the effectiveness of our pose and shape estimation method solely using event point clouds.

In summary, our contributions are as follows: (1) We are the first to tackle the challenge of 3D HMR using only event data. [1] (2) To this end, we propose EventPointMesh, a framework that treats event data as 3D spatio-temporal point clouds for mapping it to human meshes. (3) To faithfully restore human motion, we explicitly extract global and local features from event point clouds, utilizing 2D joint positions as cues for grouping. (4) Given the absence of prior methods for executing this task, we have created EPMD, a dataset for HMR comprising a vast amount of event data, intensity images, optical MoCap data, and mesh data captured under both well-lit and poorly-lit conditions;

[1] We have noticed that another research team is also attempting 3D HMR using event data as presented in an arXiv paper [14]. However, please note that this paper has not been published anywhere.

and (5) We conduct extensive experimentation and show the effectiveness of our method.

II. Related Work

A. Human Pose and Shape Estimation

Capturing human pose and shape is at the core of the evolution of VR, enabling a seamless mapping of users' natural movements into virtual spaces, thus enhancing real-time interactive experiences. This, in turn, facilitates rich content generation and animation creation, bolstering social communication within virtual environments. One of the most accurate methods to capture human pose is the optical MoCap system [1], [2], which relies on tracking markers attached to the human body using pre-installed optical sensors within a studio space. Although achieving high accuracy and frame rates, this system comes with significant constraints, including its high cost and the necessity of attaching markers to the body. Addressing these limitations, considerable advancements have been made in markerless MoCap [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], estimating 3D poses using RGB and depth cameras, substantially reducing cost and complexity. However, synchronizing and calibrating multiple camera systems still pose challenges. In contrast, the advent of deep neural networks has ushered in the proposal of 3D human pose estimation (HPE) methods based on monocular RGB cameras [5], [28], [29], [30], [31], [32], [33], [34], [35], [36]. These approaches facilitate accurate pose acquisition in scenarios where setting up pre-calibrated cameras or attaching markers may not be feasible.

However, the skeleton models primarily used in these methods, while suitable for depicting relatively simple actions like character animations [37], often require finer granularity when measuring and representing intricate human behaviors. This is because, as we interact with the world through our skin rather than internal joints, inferring body shape, contact, and gestures becomes vitally important. Against this background, the low-dimensional statistical parametric body model, SMPL [38], emerged, capable of realistically portraying human body shape and efficiently animating it. Paving the way for its application in diverse real-world scenarios such as VR/AR content creation, virtual try-ons, and computer-assisted coaching, numerous techniques have been proposed recently to estimate mesh models from images or videos [6], [7], [39], [40], [41], [42].

While these HMR methods have democratized capturing human states in various everyday contexts, they also present challenges stemming from their substantial reliance on visible light. For instance, when estimating pose using RGB(D) cameras for prolonged periods or at high frame rates [43], [44], bottlenecks in data processing and storage arise, posing significant challenges, especially for edge devices. Additionally, obtaining accurate estimations under low-light conditions, such as in dark rooms or at night, becomes challenging. Furthermore, potential privacy intrusion risks in specific use-cases cannot be overlooked. A promising solution to these challenges is employing wireless signals for pose and shape estimation [10], [45], [46], [47], [48], [49], [50]. These approaches leverage RF, including WiFi and mmWaves, known for their resilience in darkness and capability to penetrate obstacles. However, there are restrictions in using those signals near delicate electronic equipment, such as in hospitals and on aircraft, due to potential interference concerns.

Considering these challenges, we explore the use of event cameras, a modality distinct from the aforementioned methods, to recover human meshes. Unlike conventional RGB cameras that record at fixed frame rates, event cameras independently detect luminance changes per pixel, asynchronously recording only the pixels with significant changes. This mechanism achieves high temporal resolution, energy efficiency, and high dynamic range. Furthermore, since areas with no luminance changes are not recorded, event cameras present a potential for HMR in operational scenarios where RGB camera usage is restricted from a privacy standpoint [51]. Therefore, employing event cameras for HMR holds the promise of overcoming limitations encountered with RGB(D) imagery and wireless signals providing a low-light resilient, power and memory efficient, and privacy-preserving HMR technique. Details regarding pose and shape estimation techniques using event cameras will be discussed in the following subsection.

B. Human Pose and Shape Estimation Using Event Cameras

Event cameras, characterized by high temporal resolution (on the order of μ s) and a high dynamic range (over 100dB), have tremendous potential in a variety of extreme scenes. Utilizing event data has been demonstrated to be effective in many applications such as deblurring [52], scene segmentation [53], [54], visual odometry [55], corner detection [56], object recognition [57], gesture recognition [58], optical flow estimation [59], [60], depth estimation [61], Simultaneous Localization and Mapping (SLAM) [62], and autonomous driving [63]. For a comprehensive survey on event cameras, please refer to [11].

The high temporal resolution is particularly suited for capturing fast-moving objects, and in recent years, HPE and HMR using event cameras have been proposed [12], [13], [64], [65], [66], [67], [68]. In these approaches, event data is treated using various representation methods to achieve HPE and HMR. Calabrese et al. [64] proposed a method that accumulates events at consistent intervals to form "event frames" (refer to the upper part of Fig. 1(b)), and estimate 2D joint positions from those. Scarpellini et al. [65] were the first to propose a method to estimate 3D joint positions using a monocular event camera alone. This method aggregates events into synchronized tensor representations, predicts orthogonal heatmaps of each joint through a multi-layer convolutional neural network, and estimates 3D joint positions via triangulation. Zhang et al. [66] utilized a retinal-inspired event representation named TORE [69] and introduced an event camera-based 3D high-frequency HPE system called YeLan, capable of estimating dance movements in low-light conditions or against dynamic backgrounds. Chen et al. [67] successfully estimated 2D joint positions from event point clouds alone using a new event representation called "Rasterized Event Points" and a point cloud processing backbone. Shao et al. [68] tackled sparse data from low-activity body parts due to event cameras capturing only luminance changes. Using a recurrent architecture, they enhanced 2D pose estimation by modeling event frame consistency and accumulating past information. Goyal et al. [70] proposed MoveEnet, a system for 2D HPE designed for online applications, which operates at high speed using an event stream as input. By utilizing EROS [71], a representation similar to edge maps that mitigates the issues of sparsity and motion blur in event frames, they enabled pre-training of Artificial Neural Networks (ANNs) with existing large-scale image-based HPE datasets.

In addition to these skeleton-based methods, techniques have also been proposed to express human pose and shape using parametric models, enabling more detailed human state estimation. Rudnev et al. [72] proposed EventHands, the first learning-based method that enables 3D hand reconstruction from a single event stream. They utilized MANO, a mesh model specialized for the reconstruction of hand pose and shape, similar in mechanism to theSMPL, and trained their model with synthetic event streams. This approach allowed them to construct a high-speed hand reconstruction system operating at 1000Hz. Xu et al. [12] were the first to propose a model-based HMR method using event cameras. They captured event streams along with a series of grayscale images, established initial poses over time, and combined them to enable 3D HMR of fast human movements. However, this method required multiple stages of optimization and many modules for fine-grain predictions, necessitating numerous optimization hyperparameters, and as a result, demanded significant computational time in both training and inference. Zou et al. [13] trained a model to estimate mesh deformations from an initial mesh, either known or estimated from intensity images, based on compressed event frames at fixed intervals along the time axis and the inferred optical flow from them, achieving good accuracy. A common challenge among these methods is the dependency on grayscale intensity images for keyframe feature extraction and initial mesh estimation, where the mesh estimation accuracy heavily relies on the quality of the intensity images.

In this paper, we discuss a method to recover human meshes using solely event data, without relying on intensity images. Among various event data representations, such as the event frames and TORE [69], we adopt the approach that treats events as a spatiotemporal 3D point cloud, balancing processing efficiency with high temporal resolution. In the following subsection, we will elaborate on the point cloud-based method for human pose and shape estimation.

C. Human Pose and Shape Estimation Using Point Clouds

The measurement data obtained from event cameras comprises coordinates, timestamps, and polarity (direction of luminance change), forming a spatio-temporally distributed 3D point cloud. Recently, HMR methods [73], [74] utilizing Light Detection And Ranging (LiDAR) measurement data, which similarly provides 3D point cloud data composed of spatial and temporal information like event cameras, have been proposed. Moreover, methods using 3D point cloud data obtained from wireless signals for HPE [50] and HMR [10], as well as 3D HPE techniques using depth cameras [75] and approaches registering the SMPL mesh model to point cloud data acquired from 3D scans [76], [77], [78], have been proposed. Additionally, a method for estimating 2D keypoints from event point cloud [67] was proposed. Inspired by these methods, this paper adapts point cloud-based HMR approaches to a method that takes event data as input. The recent event-based HMR method [13] compressed event streams at fixed intervals, treating them as event frames, which potentially excluded significant temporal variations in pose tracking. Instead, we propose a framework treating event data as sparse point clouds. Our primary contribution is a configuration that leverages the natural characteristics of events as asynchronous signals with high temporal resolution, without depending on intensity frames, enabling mesh shape estimation for fast human movements in poorly-lit environments.

III. Method

We present EventPointMesh, a framework for estimating 3D human meshes from the spatiotemporal point clouds generated by an event camera. As shown in Fig. 2, given a sequence of event point clouds $\boldsymbol{e}^{1:T}$ segmented into T fixed-width time windows, EventPointMesh estimates the SMPL mesh sequence $\boldsymbol{m}^{1:T}$. The network is organized into four modules: the Base Module that extracts point cloud features $\boldsymbol{f}^t$ from the event point cloud $\boldsymbol{e}^t$ within the time window indexed by t; the Keypoints Module which derives a global feature vector $\boldsymbol{g}^t$ and predicts 2D human joint positions ${\boldsymbol{s}_{\text{2D}}}^t$ from $\boldsymbol{f}^t$; the Anchor Points Module designed to group the point cloud around each estimated 2D human joint positions ${\boldsymbol{s}_{\text{2D}}}^t$ and subsequently extract local features $\boldsymbol{l}^t$; and the SMPL Module that infers the SMPL mesh model from the assembled global and local feature vectors $\boldsymbol{g}^t$ and $\boldsymbol{l}^t$.

EventPointMesh adopts an approach to extract pose features from point clouds in a coarse-to-fine manner. The global features $\boldsymbol{g}^t$, extracted in the initial phase through the Keypoints Module, represent coarse features such as the subject's location, body shape, and approximate pose derived from the event point clouds. These features are utilized for mesh estimation in the subsequent SMPL Module. The 2D joint positions ${\boldsymbol{s}_{\text{2D}}}^t$, estimated by the Keypoints Module, are used in the subsequent Anchor Points Module to accurately replicate the human pose. Specifically, based on the 2D joint positions ${\boldsymbol{s}_{\text{2D}}}^t$, the event point clouds surrounding each joint are grouped, and local features $\boldsymbol{l}^t$ are extracted from each group. This process extracts fine features from events triggered by the movement of each body part, which are also utilized in the subsequent SMPL Module. Through such a coarse-to-fine feature extraction process, EventPointMesh is capable of replicating detailed human mesh reconstructions of diverse human movements.

Further elaboration on each module will be provided in the following subsections. In addition, the final subsection introduces our dataset EPMD, encompassing event data recorded under diverse lighting conditions, intensity images, optical MoCap data, and the SMPL mesh models.

A. Module Details

Base Module: This module serves as a component for converting the input event point clouds into high-dimensional features. Each time a luminance change is detected, the event camera

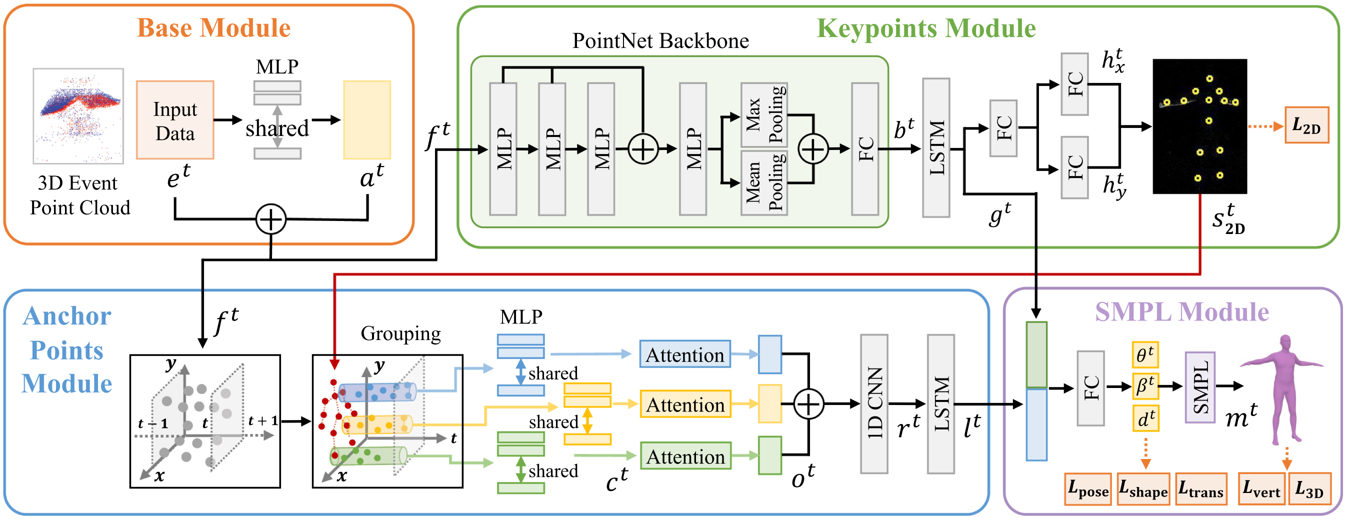

Fig. 2. Pipeline of our event-based HMR method, EventPointMesh. It consists of four modules: the Base Module, Keypoints Module, Anchor Points, and SMPL Module (Section III-A). To achieve highly accurate mesh reconstruction, EventPointMesh employs a coarse-to-fine feature extraction process: the Keypoints Module extracts global features from the event point cloud, and the AnchorPointsModule extracts local features.

generates 4D event data, which includes 2D pixel coordinates, detected timestamp, and polarity indicating the direction of the luminance change. Depending on the specifications of the event camera, the timestamp of luminance change detection is asynchronously recorded for each pixel.

Given the event point cloud segmented into fixed time windows, represented as $\boldsymbol{e}^t$, the module transforms it into high-dimensional features using a Multi-Layer Perceptron (MLP). Considering the n-th detected point within the point cloud as $\boldsymbol{e}_n^t$, this point is transformed into a high-dimensional feature vector $\boldsymbol{a}_n^t = \text{MLP}(\boldsymbol{e}_n^t; \boldsymbol{w}_a)$ by applying the MLP transformation with parameters $\boldsymbol{w}_a$. This feature vector is then concatenated with the 4D vector of $\boldsymbol{e}_n^t$, resulting in a high-dimensional vector $\boldsymbol{f}_n^t$, which is then fed into the subsequent modules. This design aims to enhance the expressiveness of the feature vectors used in subsequent processing by combining the spatiotemporal relationship of point clouds acquired during the learning process with the original attributes of each point (image coordinates, timestamp, polarity). Additionally, the preservation of the original point cloud data, along with the feature vectors, serves another purpose: the subsequent Anchor Points Module requires the spatial information of point clouds.

Keypoints Module: For accurately estimating human poses from event data, it is crucial to have a clue as to which event point corresponds to the movement of a specific body part. Hence, the Keypoints Module estimates 2D coordinates of major joints from the input point clouds. These estimated coordinates serve as reference points (hereafter referred to as “anchor points”) in subsequent modules, aiding in obtaining local features of the point clouds generated by body movements around each joint. This module is designed based on the network proposed by Chen et al. [67], considering the temporal features and the overall point cloud feature extraction.

Initially, the feature vector $\boldsymbol{f}_n^t$ of each event point cloud obtained by the Base Module is transformed into a global feature of the point cloud, $\boldsymbol{b}_n^t = \text{PointNet}(\boldsymbol{f}_n^t)$ with a network that has PointNet [79] as its backbone. Afterward, a feature vector considering temporal dynamics of the event point cloud sequence is extracted from $\boldsymbol{b}^t$ via a Bidirectional Long Short-Term Memory (BiLSTM). One of the challenges in working with event cameras is their inherent limitation: they capture only moving body parts, ignoring the stationary ones. This results in sparse data and complicates pose estimation due to the absence of complete body part information. The method by Chen et al. [67] faces challenges due to its estimation of 2D joint positions for only one frame per time window. To address this issue, we introduced a BiLSTM to construct a model that is robust against the absence of events from stationary parts, taking into account temporal information. Our method leverages features of point clouds from both forward and backward time windows, effectively compensating for these sparse areas. With the weight parameters of the BiLSTM $\boldsymbol{w}_g$, the temporal dynamics aware feature vector can be represented as $\boldsymbol{g}^{1:T} = \text{BiLSTM}(\boldsymbol{b}^{1:T}; \boldsymbol{w}_g)$, where T is the number of time windows used as input. While Chen et al. [67] utilize this feature vector for 2D joint position estimation, we view it as a spatiotemporal feature of 3D point clouds that encapsulates information about a person’s location, along with their approximate pose and body shape. Consequently, we employ it as a global feature vector for the estimation of 3D meshes in the subsequent SMPL Module.

Subsequently, the time-series feature vector $\boldsymbol{g}^t$ is transformed into two 1D heat-vectors, $(\boldsymbol{h}_x^t, \boldsymbol{h}_y^t) = \text{FC}(\boldsymbol{g}^t; \boldsymbol{w}_h)$, using fully connected layers (FC) with parameters $\boldsymbol{w}_h$. These heat-vectors represent the confidence distribution of joint positions along the x and y axes in an image plane. Chen et al. [67] introduced this representation based on the novel coordinate representation, SimCC [80]. Unlike conventional methods that estimate joint positions using 2D heatmaps, this representation allows for highly accurate estimation even in low-resolution images. The heat-vectors are denoted as $\boldsymbol{h}_x^t = [h_0, h_1, \ldots, h_i, \ldots, h_W]$ and $\boldsymbol{h}_y^t = [h_0, h_1, \ldots, h_j, \ldots, h_H]$, where i and j are the indices of the x-axis and y-axis in the image of width W and height

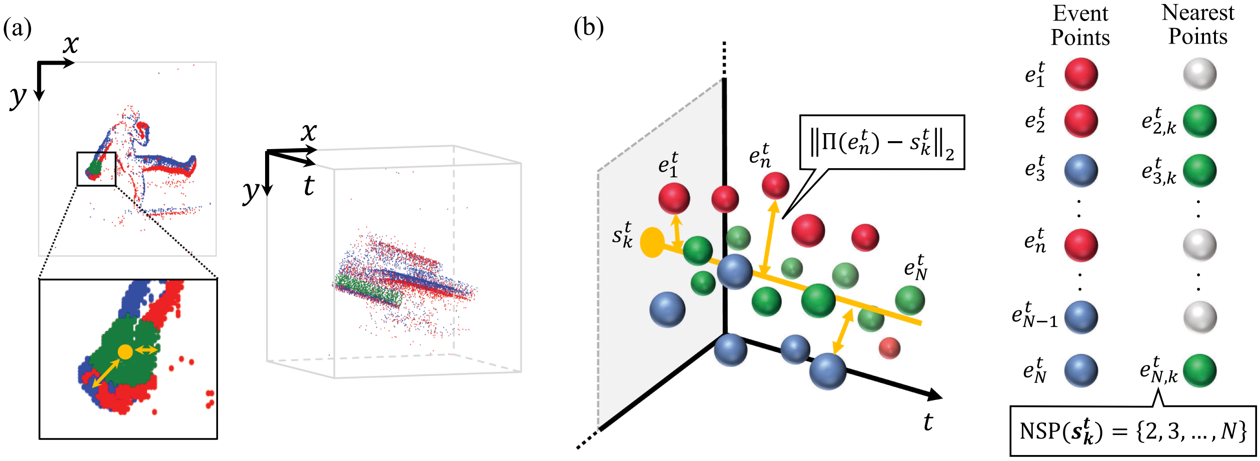

Fig. 3. Detailed illustrations of the spatiotemporal grouping of event point clouds in the Anchor Points Module. Yellow points represent the 2D joint positions estimated by the Keypoints Module, while red and blue indicate the input event point clouds, and green represents the grouped points. (a) An example of applying spatiotemporal grouping to real event data, focusing on the right wrist, though in practice, this process is applied individually to all major joints. (b) A detailed view of the nearest points selection. For each point in the event point clouds scattered in space and time, the L2 norm on the image plane with the 2D joint coordinates is calculated, and the nearest G points with the closest distances are selected as a group.

H, respectively. The 2D joint positions $\boldsymbol{s}_{\mathrm{2D}}^t = (\boldsymbol{s}_x^t, \boldsymbol{s}_y^t)$ are then determined by finding the coordinates with the maximum value in each heat-vector. Specifically, the coordinates are obtained by $\boldsymbol{s}_x^t = \arg\max_i(\boldsymbol{h}_x^t(i))$ and $\boldsymbol{s}_y^t = \arg\max_j(\boldsymbol{h}_y^t(j))$.

Anchor Points Module: In order to capture the detailed features of the point cloud and to achieve accurate mesh estimation, we utilize anchor points, which are the 2D joint coordinates estimated by the Keypoints Module. These points facilitate the grouping of event point clouds scattered across the 3D spatiotemporal space. This approach enables us to classify events stemming from body movements around each joint and extract localized features. This network configuration is inspired by mmMesh, proposed by Xue et al. [10], which estimates 3D human meshes from millimeter-wave point clouds. Whereas mmMesh utilized fixed-size 3D grid-based anchor points within an xyz spatial framework, our approach adopts 2D joint points as anchor points, incorporating the temporal dimension into the xy image plane. These 2D joint positions are then used to group surrounding event points in a 3D spatiotemporal space, based on their euclidean distance in 2D image coordinates. The details of this process are illustrated in Fig. 3. For clarity in notation: the coordinates of the k-th anchor point at the time window t are denoted as $\boldsymbol{s}_k^t \in \mathbb{R}^2$. The set of indices of the grouped nearest G points is denoted as $\mathrm{NSP}(\boldsymbol{s}_k^t)$. A grouped event point is referred to as $\boldsymbol{e}_{n,k}^t$ where $n \in \mathrm{NSP}(\boldsymbol{s}_k^t)$, and its 2D coordinates on the image plane are given by a function Π as $\Pi(\boldsymbol{e}_{n,k}^t) \in \mathbb{R}^2$. Using these definitions, the feature vector $\boldsymbol{c}_{n,k}^t$ is derived through an MLP operation as $\boldsymbol{c}_{n,k}^t = \mathrm{MLP}([\boldsymbol{s}_k^t, \|\Pi(\boldsymbol{e}_{n,k}^t) - \boldsymbol{s}_k^t\|_2, \boldsymbol{f}_n^t]; \boldsymbol{w}_c)$. The input to the MLP is a vector formed by concatenating the anchor point coordinates, the distance between the grouped points and the anchor point, and the feature vector from the Base Module. This design enhances our model's capability to recognize the spatiotemporal relationships between anchor points and their neighboring points.

The feature vector $\boldsymbol{c}_n^t$ is then fed into the attention network to extract particularly important event point features from the grouped point cloud. In existing point cloud based approaches, such as PointNet [79], the Max Pooling operation is typically used to extract the most prominent features by eliminating redundant information. However, as the point cloud grouped by this module already contains less redundant information, we have adopted an attention mechanism to aggregate the features of all points in the group while preventing the loss of important information due to the Max Pooling operation. Representing the attention network as a mapping function ATTN with weight parameters $\boldsymbol{w}_o$, the feature vector $\boldsymbol{c}_n^t$ of each grouped point is transformed into the following feature vector: $\boldsymbol{o}_k^t = \sum_{n\in\mathrm{NSP}(\boldsymbol{s}_k^t)} \mathrm{ATTN}(\boldsymbol{c}_n^t; \boldsymbol{w}_o) \cdot \boldsymbol{c}_n^t$. Furthermore, to aggregate the feature vectors $\boldsymbol{o}_k^t$ of each anchor point and transform them into features considering the spatial relationship between each group, i.e., the events caused by each body joint, we introduce a 1D convolution layer and obtain the feature vector $\boldsymbol{r}^t = \mathrm{1DCNN}(\boldsymbol{o}^t; \boldsymbol{w}_r)$. This $\boldsymbol{r}^t$ is transformed into a temporal relationship aware feature vector by using BiLSTM, similar to the Keypoints Module. Letting $\boldsymbol{w}_l$ be the weight parameters of BiLSTM, the module obtains final local feature vector $\boldsymbol{l}^{1:T} = \mathrm{BiLSTM}(\boldsymbol{r}^{1:T}; \boldsymbol{w}_l)$.

SMPL Module: The primary role of the SMPL Module is to estimate the SMPL mesh model [38] based on the global and local features of the event point clouds obtained from the modules described above. By feeding these features into an MLP, we derive vectors representing a human pose, shape, and translation. Subsequently, by using the SMPL regression model [38], we estimate mesh vertices and 3D human joint positions. The SMPL model has gained traction in contemporary research on pose and shape estimation. Its strength lies in its ability to represent the complex and diverse poses and shapes of the human body using a limited number of parameters. The model parameters encompass pose parameters $\boldsymbol{\theta} \in \mathbb{R}^{24\times 3}$, indicating the relative rotations of 23 joints and the global rotation of the root joint, and shape parameters $\boldsymbol{\beta} \in \mathbb{R}^{10}$, which influence aspects like height, weight, and limb ratios. Based on these parameters, the SMPL regression model estimates the triangular mesh vertices, denoted as $\boldsymbol{v} \in \mathbb{R}^{6890}$, and the 3D joint positions, represented by $\boldsymbol{s}_{\text{3D}} \in \mathbb{R}^{24\times 3}$. Additionally, the global translation of a person can be captured using a parameter $\boldsymbol{d} \in \mathbb{R}^3$. The parameter indicates the location information for positioning the SMPL model in 3D space, specifically denoting the 3D coordinates of the root joint.

Delving into the technical details, the feature vectors $\boldsymbol{g}^t$ and $\boldsymbol{l}^t$, obtained from the Keypoint and Anchor Points Modules respectively, are concatenated. This combined feature vector is then transformed into SMPL parameters using FC layers, as represented by the equation $[\boldsymbol{\theta}^t, \boldsymbol{\beta}^t, \boldsymbol{d}^t] = \text{FC}([\boldsymbol{g}^t, \boldsymbol{l}^t]; \boldsymbol{w}_p)$, where $\boldsymbol{w}_p$ represents the weight parameters of the MLP. The mesh $\boldsymbol{m}^t = (\boldsymbol{v}^t, {\boldsymbol{s}_{\text{3D}}}^t)$ is subsequently derived from $\boldsymbol{\theta}^t, \boldsymbol{\beta}^t, \boldsymbol{d}^t$ using the pre-trained SMPL regression model [38], as $\boldsymbol{m}^t = \text{SMPL}(\boldsymbol{\theta}^t, \boldsymbol{\beta}^t) + \boldsymbol{d}^t$. To ensure the pre-trained model's integrity, which uniquely determines the mesh model from the pose and shape parameters, its weights are kept fixed during both the training and testing phases.

B. Loss Function

Our EventPointMesh network is trained using a comprehensive loss function designed to address key aspects such as pose, shape, translation, mesh vertex, and 3D joint positions.

Pose Loss ($\mathcal{L}_{\text{pose}}$) measures the Mean Squared Error (MSE) between the predicted pose vector $\boldsymbol{\theta}_k^t$ and ground truth $\hat{\boldsymbol{\theta}}_k^t$ across all time-windows T and joints K.

$$\mathcal{L}_{\text{pose}} = \frac{1}{TK} \sum_{t=1}^{T} \sum_{k=1}^{K} \|\boldsymbol{\theta}_k^t - \hat{\boldsymbol{\theta}}_k^t\|_2^2, \tag{1}$$

Shape Loss ($\mathcal{L}_{\text{shape}}$) mesures the MSE between the estimated shape parameter $\boldsymbol{\beta}^t$ and its corresponding ground truth $\hat{\boldsymbol{\beta}}^t$ over all time-windows.

$$\mathcal{L}_{\text{shape}} = \frac{1}{T} \sum_{t=1}^{T} \|\boldsymbol{\beta}^t - \hat{\boldsymbol{\beta}}^t\|_2^2, \tag{2}$$

Translation Loss ($\mathcal{L}_{\text{trans}}$) ensures accurate translation prediction by calculating the MSE between the predicted translation vector $\boldsymbol{d}^t$ and ground truth $\hat{\boldsymbol{d}}^t$.

$$\mathcal{L}_{\text{trans}} = \frac{1}{T} \sum_{t=1}^{T} \|\boldsymbol{d}^t - \hat{\boldsymbol{d}}^t\|_2^2, \tag{3}$$

Vertex Loss ($\mathcal{L}_{\text{vert}}$) aims to optimize the prediction of vertices by measuring the MSE of predicted vertices $\boldsymbol{v}^t$ with the ground truth $\hat{\boldsymbol{v}}^t$ across all time-windows and vertices P.

$$\mathcal{L}_{\text{vert}} = \frac{1}{TP} \sum_{t=1}^{T} \sum_{p=1}^{P} \|\boldsymbol{v}^t - \hat{\boldsymbol{v}}^t\|_2^2, \tag{4}$$

3D Joint Loss ($\mathcal{L}_{\text{3D}}$) ensures accurate 3D joint position prediction by calculating the MSE of the predicted 3D joint positions ${\boldsymbol{s}_{\text{3D}}}^t$ with their respective ground truth ${\hat{\boldsymbol{s}}_{\text{3D}}}^t$.

TABLE I
DATASETS FOR EVENT-BASED HUMAN POSE AND SHAPE ESTIMATION

Dataset	Seq/Sub	Frame	Pose	Shape	Poorly-lit Env.
DHP19 [64]	33/17	87k	Yes	No	No
Yelan-Syn [66]	8/10	4M	Yes	No	Yes
Yelan-Real [66]	4/9	446k	Yes	No	Yes
EventCap [12]	2/6	N/A	N/A	N/A	N/A
MMHPSD [13]	12/15	240k	Yes	Yes	No
EPMD (Ours)	16/3	228k	Yes	Yes	Yes

$$\mathcal{L}_{\text{3D}} = \frac{1}{TK} \sum_{t=1}^{T} \sum_{k=1}^{K} \|{\boldsymbol{s}_{\text{3D}}}^t - {\hat{\boldsymbol{s}}_{\text{3D}}}^t\|_2^2, \tag{5}$$

2D Joint Loss ($\mathcal{L}_{\text{2D}}$) measures the discrepancy between the predicted heat-vector $\boldsymbol{h}^t$ and its ground truth $\hat{\boldsymbol{h}}^t$. To compute this loss, the ground truth 3D joint positions $\hat{\boldsymbol{s}}_{\text{3D}}$ are first projected onto the 2D image plane. Using known camera parameters and the projection function π, the 2D joint coordinates $\hat{\boldsymbol{s}}_{\text{2D}} = (\hat{s}_x, \hat{s}_y)$ are obtained by $\hat{\boldsymbol{s}}_{\text{2D}} = \pi(\hat{\boldsymbol{s}}_{\text{3D}})$. Following the projection, a Gaussian filter is applied to generate the heat-vectors, which are represented as $\hat{\boldsymbol{h}}_x = [\hat{h}_0, \hat{h}_1, \ldots, \hat{h}_i, \ldots, \hat{h}_W]$ and $\hat{\boldsymbol{h}}_y = [\hat{h}_0, \hat{h}_1, \ldots, \hat{h}_j, \ldots, \hat{h}_H]$. The confidence values for $\hat{\boldsymbol{h}}_x$ are calculated as:

$$\hat{h}_i = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(i - \hat{s}_x)^2}{2\sigma^2}\right). \tag{6}$$

The confidence values for $\hat{\boldsymbol{h}}_y$ can be computed similarly by replacing i with j and $\hat{s}_x$ with $\hat{s}_y$. To quantify the divergence between the predicted and ground truth heat-vectors, the Kullback–Leibler (KL) divergence is employed:

$$\mathcal{L}_{\text{2D}} = \frac{1}{T} \sum_{t=1}^{T} \text{KL}(\boldsymbol{h}^t \parallel \hat{\boldsymbol{h}}^t). \tag{7}$$

Overall Loss ($\mathcal{L}$) aggregates the aforementioned losses to train our EventPointMesh network and is described by:

$$\begin{aligned} \mathcal{L} = {} & \lambda_{\text{pose}}\mathcal{L}_{\text{pose}} + \lambda_{\text{shape}}\mathcal{L}_{\text{shape}} + \lambda_{\text{trans}}\mathcal{L}_{\text{trans}} \\ & + \lambda_{\text{vert}}\mathcal{L}_{\text{vert}} + \lambda_{\text{3D}}\mathcal{L}_{\text{3D}} + \lambda_{\text{2D}}\mathcal{L}_{\text{2D}}. \end{aligned} \tag{8}$$

In this formulation, the λ coefficients serve as hyperparameters that balance the impact of each loss component during the model's training phase. Notably, while mesh vertices $\boldsymbol{v}$ and 3D joint positions $\boldsymbol{s}_{\text{3D}}$ are uniquely determined by the pose parameters $\boldsymbol{\theta}$ and shape paremeter $\boldsymbol{\beta}$, the vertex loss $\mathcal{L}_{\text{vert}}$ and the 3D joint loss $\mathcal{L}_{\text{3D}}$ are included to enhance training stability and accelerate convergence.

C. EventPointMesh Dataset (EPMD)

Why the Original Dataset is Required? We propose our original dataset EPMD to address the lack of a dataset for HMR in low-light conditions, a domain where event cameras are particularly advantageous. Table I provides a comparison of EPMD with existing datasets for HPE and HMR, focusing

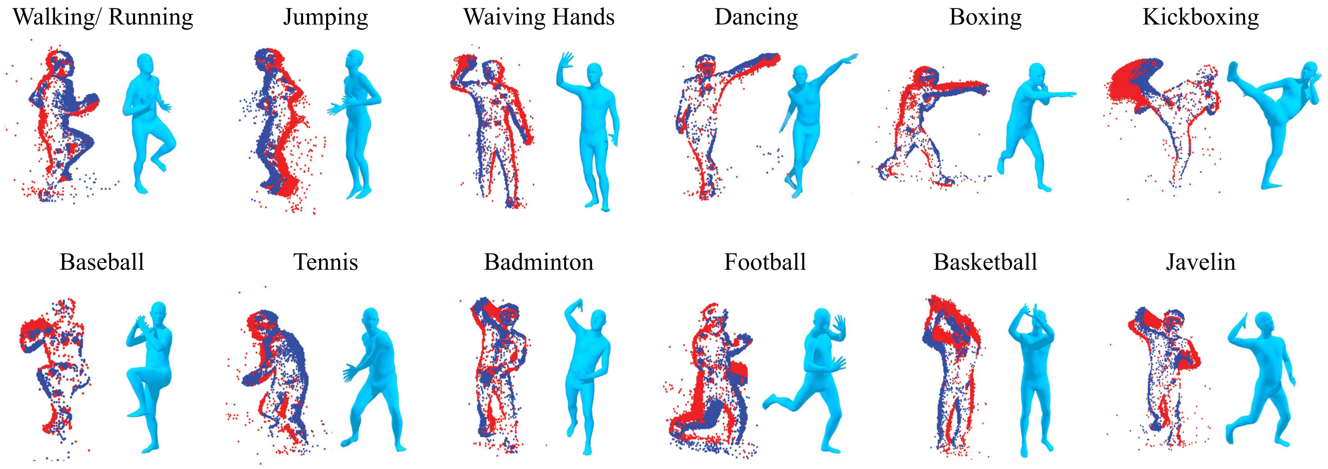

Fig. 4. The EventPointMesh Dataset (EPMD) encompasses diverse actions from daily activities to sports motions. Subjects performed the actions multiple times in both well-lit and poorly-lit environments. Captured using two event cameras and a MoCap system, EPMD comprises synchronized event data, intensity images, SMPL meshes, and optical MoCap data.

on metrics such as the number of sequences per subject, total subjects, total frame count, as well as the availability of data on poses, shapes, and poorly-lit environments. EPMD presents mesh data acquired in varied lighting conditions, a feature not available in other HMR datasets such as EventCap [12] and MMHPSD [13]. This enables the exploration of HMR in lighting conditions where conventional frame-based cameras struggle, highlighting the capabilities of event cameras. While the number of subjects in our dataset is fewer than existing datasets, EPMD offers a comparable volume of captured data ranging from daily activities to sports motions, as illustrated in Fig. 4. This supports the creation of adaptable and broadly applicable HMR models, in contrast to datasets like Yelan-Syn and Yelan-Real [66], which are limited to dance movements. The dataset comprises not only mesh data but also synchronized intensity images, event data, and optical MoCap marker data. This multifaceted composition ensures EPMD is a valuable resource for advancing HMR research, particularly in challenging lighting conditions.

Data Acquisition: EPMD was acquired using two event cameras and an optical MoCap system. We used iniVation DAVIS346 models [81] with a spatial resolution of 346 × 260 pixels and a temporal resolution of 1 μs as event cameras. These cameras recorded synchronized grayscale frame images and event data, which were transmitted in packets. The event data within these packets were then converted into point cloud data segmented at fixed time intervals for input into the network. The event point clouds contain an average of 270 k points per second across the entire dataset. The OptiTrack Motive [1] is used as the optical MoCap system. Three subjects were asked to wear a MoCap suit with markers and were recorded performing various motions. They performed medium-speed motions like walking, jogging, jumping, waving hands, and kicking, as well as fast motions such as fast running, dancing, boxing, kickboxing, baseball, tennis, badminton, football, basketball, and javelin (Fig. 4). These actions are captured in both bright and dark conditions. Our dataset includes approximately 228 k intensity frames shot at about 15 fps in bright conditions and about 4 fps in dark conditions. It also contains event data synchronized with the intensity images, MoCap marker trajectory data captured at 120 fps, and data from the generated SMPL mesh model, totaling about 3 hours. The study was formally approved by the Bioethics Committee of the Graduate School of Science and Technology at Keio University under approval number 2023-112.

Annotation: The annotation of the SMPL mesh model is conducted based on the trajectory data of the MoCap markers. By labeling the markers with their corresponding position names during recording with the OptiTrack, it becomes feasible to employ the Mosh++ [82], [83] method, which fits the SMPL model to the labeled marker data through optimization. The SMPL model generated by Mosh++ contains pose parameters, shape parameters, vertex coordinates of the mesh model, 3D joint positions, and 3D translation data in the MoCap coordinate system. A total of 24 joints are annotated, which include: Pelvis, Left and Right Hips, Left and Right Knees, Left and Right Ankles, Left and Right Feet, Spine1, Spine2, Spine3, Neck, Head, Left and Right Collars, Left and Right Shoulders, Left and Right Elbows, Left and Right Wrists, and Left and Right Hands. The ground truth data of 2D joint positions used in the Keypoints Module is acquired by projecting the 3D joint positions onto the image in the event camera coordinate system. The intrinsic camera parameters used in this projection are obtained through calibration using a checkerboard pattern. The extrinsic parameters are calculated by obtaining corresponding points in the MoCap and event camera coordinate systems and solving the Perspective-n-Point (PnP) problem.

IV. Experiments

A. Experimental Settings

Dataset: To demonstrate the efficacy of our approach, we employed both the existing large-scale dataset MMHPSD [13] and our EPMD for experiments. The MMHPSD comprises data from a total of 15 subjects: 11 males and 4 females. Each subject performed a total of 21 different motions, categorized into three groups based on speed: fast, middle, and slow. Each motion was repeated four times. For each subject, 12 video clips (totaling 180

TABLE II
QUANTITATIVE RESULTS FROM EXPERIMENTS USING THE MMHPSD AND EPMD DATASETS NOTE: FOR DETAILS ON THE NOTATION OF VALUES IN THIS TABLE, PLEASE REFER TO SECTION IV-B 'BASELINE METHOD'

Dataset	Method	MPJPE [mm] (↓)	PEL-MPJPE [mm] (↓)	PA-MPJPE [mm] (↓)	PVE [mm] (↓)	PCKh@0.5 [%] (↑)	Miss Rate [%] (↓)
MMHPSD (All-Subject)	EventHPE(GT)	72.3	52.6	41.3	50.5	86.0	-
	EventHPE(VIBE)	-	71.3	51.4	(70.5)	79.6	1.4
	EPM(Ours)	97.4	**59.1**	**42.8**	**51.9**	**84.8**	**0.0**
MMHPSD (Cross-Subject)	EventHPE(GT)	103.9	80.4	64.8	79.0	76.0	-
	EventHPE(VIBE)	-	88.5	66.6	(86.0)	69.3	1.4
	EPM(Ours)	145.1	**86.3**	**61.6**	**74.7**	**71.9**	**0.0**
EPMD (All-Subject)	EventHPE(GT)	198.0	90.3	71.2	81.1	70.2	-
	EventHPE(VIBE)	-	128.9	95.7	(109.0)	55.0	2.1
	EPM(Ours)	110.7	**86.0**	**60.7**	**71.8**	**71.7**	**0.0**

clips) were collected, each consisting of approximately 1,300 frames spanning about a minute and a half at 15 fps. Additionally, our EPMD uniquely contains a rich set of data recorded in low-light conditions, which are not found in existing datasets for HMR. Therefore, we used the EPMD to assess the low-light robustness of our proposed method.

Implementation Details: During training, the event data was segmented by time windows corresponding to the intervals of the intensity image frames and treated as point cloud data between adjacent frames. Given the vast volume of raw data points, which posed challenges in storage and data processing, the point cloud within each time window was randomly sampled down to 7,500 points to reduce data size. The sequence of event data fed into the network was time series point cloud data for 15 consecutive time windows. In the Keypoints Module, we identified 13 primary human body joints (Pelvis, Left and Right Knee, Left and Right Ankle, Spine, Head, Left and Right Shoulder, Left and Right Elbow, and Left and Right Wrist) and estimated their 2D joint positions following Xue et al. [10]. In the Anchor Points Module, we grouped the nearest 500 points from the spatiotemporal domain based on the 2D euclidean distance on the image plane, centered around the 2D joint coordinates. For network training, we employed the Adam optimizer [84]. The weights for each element in the loss function, namely λ_{pose}, λ_{shape}, λ_{trans}, λ_{vert}, λ_{3D}, and λ_{2D}, were empirically determined to be 10, 10, 10, 100, 10, and 1, respectively. To discourage settling into local minima during the learning process, the learning rates for the Base Module and Keypoints Module were set to 1e-4, while that of the Anchor Points Module and SMPL Module were set to 1e-3, based on experimental findings. The batch size was set to 8. End-to-end learning was conducted across the entire network, including all modules.

Evaluation Metrics: Referring to the prior research [13], we evaluated experimental results using six metrics: mean per joint position error (MPJPE), pelvis-aligned MPJPE (PEL-MPJPE), Procrustes-aligned MPJPE (PA-MPJPE), per vertex error (PVE), percentage of correct keypoints (PCKh@0.5), and miss rate. In Tables II and IV, MPJPE, PA-MPJPE, PEL-MPJPE, and PVE are all denoted in mm.

MPJPE calculates the average 3D euclidean distance between the ground truth and estimated joint positions. PEL-MPJPE computes MPJPE after aligning the translation of the root joint. PA-MPJPE is an evaluation metric that calculates MPJPE after applying a Procrustes analysis [85], which aligns the predicted and ground true poses by minimizing the distances between corresponding joints through translation, scaling, and rotation adjustments. PVE metric assesses the 3D euclidean distance between the true and estimated coordinates of each vertex of the SMPL mesh model. MPJPE and PVE are respectively computed using the following equations.

$$\mathbf{E}_{\text{MPJPE}} = \frac{1}{TK}\sum_{t=1}^{T}\sum_{k=1}^{K}\|\boldsymbol{s}_{\text{3D}}{}_t^k - \hat{\boldsymbol{s}_{\text{3D}}}{}_t^k\|_2, \tag{9}$$

$$\mathbf{E}_{\text{PVE}} = \frac{1}{TP}\sum_{t=1}^{T}\sum_{p=1}^{P}\|\boldsymbol{q}_t^p - \hat{\boldsymbol{q}}_t^p\|_2, \tag{10}$$

In the above equation, T, K, P, $\boldsymbol{s}_{\text{3D}}$, and $\boldsymbol{q}$ represent the number of time windows for the event point clouds, the number of body joints, the number of vertices in the SMPL mesh model, the 3D coordinates of the joints, and the vertex coordinates of the SMPL mesh model, respectively. $\hat{\boldsymbol{s}}_{\text{3D}}$ and $\hat{\boldsymbol{q}}$ represent the ground truth corresponding to the estimated values $\boldsymbol{s}_{\text{3D}}$ and $\boldsymbol{q}$. PCKh@0.5 evaluates the percentage of joints whose euclidean distance between the ground truth and the estimated position is less than 50% of the bone length from the neck to the head. Miss rate evaluates the percentage of failure to output a mesh in the test data.

B. Comparison Against Event-Based Methods

Through three experiments utilizing two distinct datasets, MMHPSD and EPMD, we demonstrate the efficacy of our approach in estimating meshes solely from event data compared to baseline methods. The experiments were conducted under two different settings: (i) an all-subject setting, where data for all subjects was divided into training and test sets, and (ii) a cross-subject setting, where different subjects' data were used for training and testing. Given the large number of subjects included in the MMHPSD, the experiment was conducted under both conditions (i) and (ii). On the other hand, EPMD, designed for assessing performance under diverse lighting conditions, contains fewer subjects; hence, only condition (i) was applied in

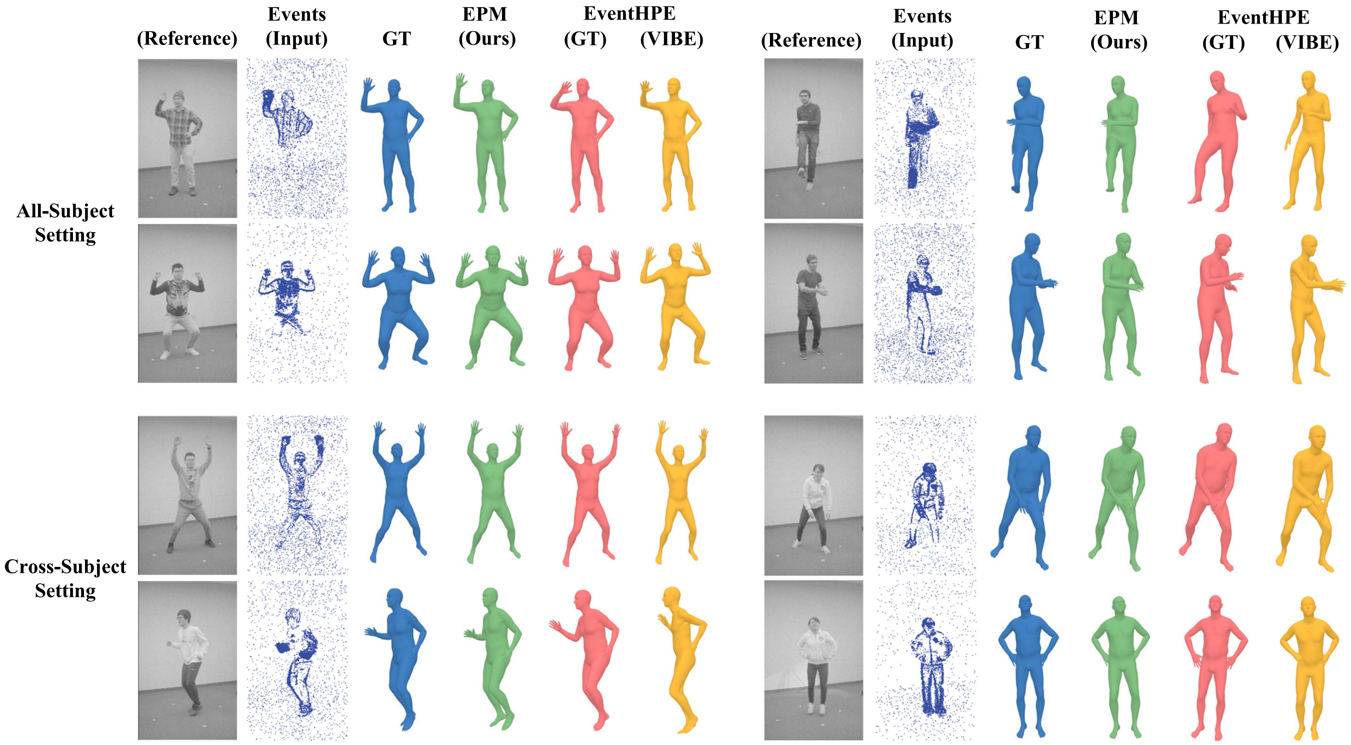

Fig. 5. Qualitative results from the experiment using MMHPSD as detailed in Section IV-B. The top rows display the results under the all-subject setting, while the bottom rows show the results under the cross-subject setting. Although our proposed method only uses event data as input, it estimates the mesh with an accuracy comparable to EventHPE(VIBE), which uses intensity images to estimate the initial mesh for each sequence, and EventHPE(GT), which employs ground truth mesh for the initial frame of the sequences.

its experiments. This subsection first details the baseline methods, followed by a description of the experiments conducted using each dataset.

Baseline Methods: Our approach is the first to estimate the 3D human meshes solely from the point cloud of event data, and no existing methods are tackling the same task. Therefore, we compared our network model against EventHPE [13], a similar approach to ours. EventHPE differs significantly from our method in that it requires intensity images in addition to event frames to obtain the initial mesh model at the starting point for the sequence. Subsequent mesh models in the sequence are then estimated based on their displacements from this initial model. EventHPE computes this initial mesh model by using VIBE [7], which estimates the SMPL mesh from videos.

Throughout the experiments, EventHPE using VIBE is denoted as EventHPE(VIBE), and the method using the ground truth mesh as the initial mesh of the sequence is referred to as EventHPE(GT). Following the experimental setup of Zou et al. [13], EventHPE(GT) operates under the assumption that the mesh of the initial frame in each sequence is known, and the sequence consists of 8 frames. This method represents the upper bound of estimation accuracy that EventHPE network can achieve. In Table II, the highest accuracy values, when including EventHPE(GT), are underlined, while the best scores excluding it are highlighted in bold. The VIBE [7] estimates camera parameters based on a weak perspective projection model, not accounting for global translation, and thus we do not report MPJPE for EventHPE(VIBE). Additionally, the mesh model used by VIBE is a gender-neutral model, unlike the gender-specific models used in MMHPSD and EPMD. Consequently, the PVE values for EventHPE(VIBE) are provided in parentheses as reference values.

Evaluation on MMHPSD (All-Subject): We evaluated the generalization ability of the models under the all-subject setting. Specifically, the networks were trained using three out of the four video sequences representing each subject's actions and used the remaining sequence for testing. The qualitative and quantitative results are presented in Table II and the upper part of Fig. 5, respectively. A comparison of the methods, as shown in Fig. 5, reveals that despite slight discrepancies in body orientation and limb positions, the models generally achieve mesh estimations of high accuracy that closely mirror the ground truth. A detailed examination of Table II, which presents results across the entire test dataset, indicates that EventHPE(GT) with ground truth-based initialization achieves the highest accuracy. Without EventHPE(GT), our method EPM(Ours) emerged as the most accurate. This underscores that our approach, despite not relying on intensity images, outperforms existing methods that estimate meshes from intensity images and event data.

Evaluation on MMHPSD (Cross-Subject): We aimed to assess the generalization performance of the models across different subjects. We conducted a cross-subject evaluation by training our network on data from 12 out of the 15 subjects and employing the remaining 3 subjects' data for testing. The qualitative and quantitative results are shown in Table II and the lower part of Fig. 5, respectively. Our method consistently surpassed

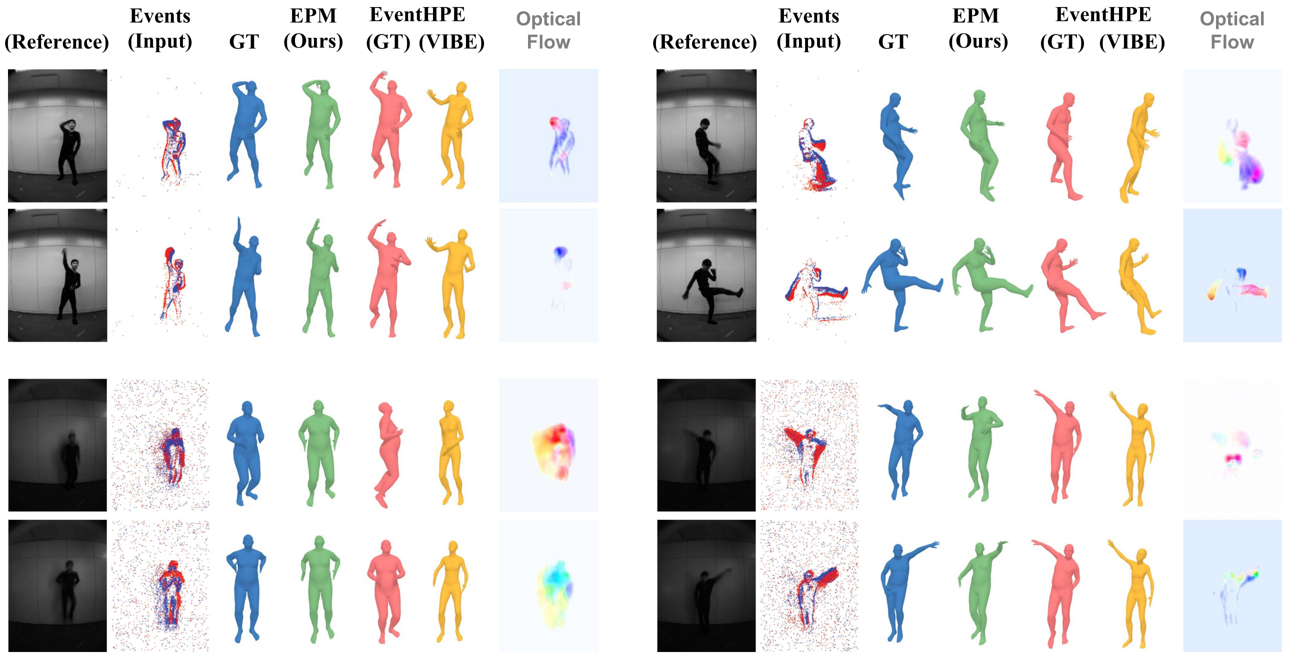

Fig. 6. Qualitative results from the experiment using EPMD under the all-subject setting (Section IV-B). The top two rows display well-lit data results, while the bottom two rows show poorly-lit data results. The images, segmented into four blocks separated by wide spaces, are adjacent frames arranged in the top and bottom rows. Note that the Optical Flow, assisting mesh estimation in EventHPE(VIBE) and EventHPE(GT), comes from the FlowNet module and is unrelated to EMP (Ours). Our method estimates a mesh closer to ground truth than EventHPE, and is significantly better than them, especially for data in poorly-lit environments.

EventHPE(VIBE) in all metrics, and the results for PEL-MPJPE and PVE were even better than those for EventHPE(GT). PEL-MPJPE and PVE evaluate the error in 3D joint coordinates when aligning translation and rotation of the root joint. These outcomes suggest that our method excels in estimating the pose and shape relative to the root joint from event data compared to that of EventHPE [13]. However, the influence of root joint rotation on accuracy was evident, indicating room for improvement.

Comparing the outcomes for the all-subject setting and the cross-subject setting revealed that all metrics degraded in the cross-subject setting, hinting at a reduced estimation accuracy when predicting non-included individuals. This suggests that variations in the same movements by different subjects, along with differences in clothing and physique, led to distinct event data characteristics. Future work could focus on developing techniques to mitigate the effects of clothing and physique variations on data characteristics.

Evaluation on EPMD (All-Subject): To evaluate the robustness of the methods against variations in lighting, we conducted an all-subject evaluation using EPMD that contains data captured while subjects performed various medium to high-speed movements in both well-lit and poorly-lit environments. Due to the limited number of subjects, in this experiment, we utilized 80% of the takes from each subject's data for training and the remaining 20% for testing. The experimental results are presented in Fig. 6 and Table II. The optical flows, displayed in Fig. 6, were estimated by the FlowNet module used in EventHPE. It is important to note that in our experiments, this optical flow is commonly utilized in both EventHPE(GT) and EventHPE(VIBE). In contrast, our proposed method, EPM(Ours), does not make use of optical flow.

As can be seen from Table II, our method achieved the highest accuracy across all metrics, even when including EventHPE(GT). While the EventHPE [13] inputs not only the event frames but also the optical flow estimated from them into the mesh estimation network ShapeNet, its performance tends to deteriorate when the optical flow estimation is not accurate, as illustrated in the lower part of Fig. 6. In particular, in the bottom right two rows showcasing the EventHPE method, both adjacent frames appear to be in almost the same pose. This is due to EventHPE's mechanism of using a sequence of data comprising optical flow and event frames from eight adjacent frames for mesh estimation. If the sequence includes an optical flow that has significantly failed in estimation, it greatly impacts the accuracy of subsequent frame estimations, resulting in a failure to replicate changes in the subject's movement.

One potential cause for these inaccuracies in optical flow estimation lies in the presence of motion blur in the intensity images and the reduced frame rates. When capturing data in dim lighting, the camera requires longer exposure times to gather sufficient light, leading not only to motion blur but also to reduced frame rates. This reduction in frame rate means that the poses between adjacent frames can vary significantly. Since EventHPE trains its FlowNet to predict optical flow based on such intensity images, both the motion blur and the large pose differences between frames can hinder accurate optical flow estimation. This adverse effect impacts not just the EventHPE [13] network but also VIBE [7], which is utilized to predict the mesh of the sequence's initial frame. In contrast, EventPointMesh can estimate meshes with high accuracy without being affected by issues related to intensity images.

The miss rate in Table II indicates the proportion of test data frames where the mesh was not estimated. EventHPE(VIBE) failed to estimate the mesh for 1.4% of the frames in MMHPSD and 2.1% in EPMD, resulting in missing outputs. In contrast, our method, which solely relies on event data without using intensity images, achieves the miss rate of 0.0% as it is not subject to the adverse effects of the intensity image caused by the dark environment. The miss rate of 0.0% represents an absolute zero value, indicating that our proposed method reliably outputs a mesh model based on the input event data. The presence of missing parts in the estimated mesh could necessitate additional processing, such as motion infilling, to fill in the gaps, particularly in applications requiring online streaming. This could potentially affect real-time performance. From this perspective, the error rate of 0.0% for our proposed method is an important and beneficial metric for the development of future applications.

C. Comparison Against a Frame-Based Method

We demonstrate the effectiveness of our method utilizing an event camera under conditions that are challenging for traditional frame-based cameras, namely dim environments where images tend to become dark, unclear, and prone to motion blur. To this end, we conduct accuracy comparison experiments between the existing frame-based camera method, VIBE [7], and our proposed method, EventPointMesh. We divided the EPMD dataset into data captured in light and dark environments and tested both VIBE and EventPointMesh against each setting. The event camera used in this study dynamically adjusts exposure time according to lighting conditions to obtain images with sufficient brightness. During our data collection, the camera captured at approximately 15 fps in light environments and around 4 fps in dark environments. Consequently, the number of frames was 188k in the light environment and 40 k in the dark environment, leading to an imbalance in the number of frames between the two conditions. However, since the subjects performed the same actions for the same duration under both lighting conditions, the data is balanced in terms of the types and frequency of actions performed by the subjects across both conditions. Tests conducted in light environments are denoted as VIBE(L) and EPM(L), while those in dark environments are marked as VIBE(D) and EPM(D). Tests on data from both lighting environments combined are labeled as VIBE(L+D) and EPM(L+D). The results of these experiments are presented in Table III. Additionally, qualitative results are shown in Fig. 7.

TABLE III
QUANTITATIVE RESULTS OF THE COMPARISON BETWEEN EVENT-BASED AND FRAME-BASED METHODS

Method	PEL-MPJPE [mm] (↓)	PA-MPJPE [mm] (↓)	PVE [mm] (↓)
VIBE(L)	150.4	75.6	89.7
VIBE(D)	212.9	123.8	145.7
VIBE(L+D)	158.9	82.1	97.3
EPM(L)	84.4	59.4	70.0
EPM(D)	103.3	75.0	90.2
EPM(L+D)	86.0	60.7	71.8

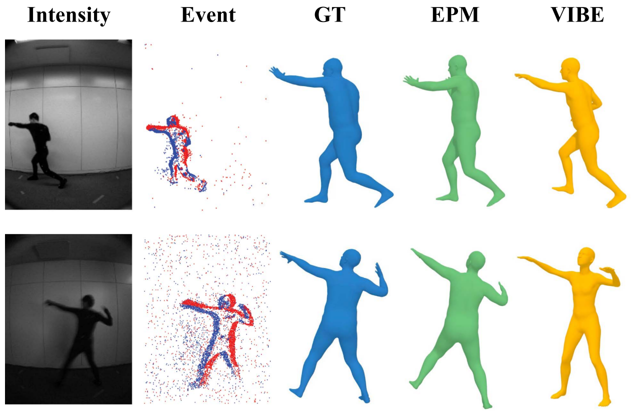

Fig. 7. Qualitative results of the experiment comparing our event-based method with the frame-based method. The top row displays the mesh estimation results of both methods in bright environments, while the bottom row presents the estimation results in dark environments. In the bottom row, it is shown that the frame-based VIBE incorrectly reverses the orientation of the estimated meshes, whereas EventPointMesh demonstrates robustness in dark environments, accurately estimating the correct mesh orientation.

The results from Table III demonstrate that our proposed method achieves higher accuracy across all conditions, whether in light, dark, or mixed environments. Furthermore, comparing the relative increase in error from light (L) to dark (D) environments between VIBE and EPM reveals that VIBE experiences a larger increase in error. For instance, while EPM shows an increase of +22.4% in PEL-MPJPE and +26.3% in PA-MPJPE, VIBE's errors escalate to +41.6% in PEL-MPJPE and +63.8% in PA-MPJPE. This indicates that the frame-based method VIBE is more adversely affected by dim lighting. Moreover, comparing the increase in the magnitude of PEL-MPJPE relative to PA-MPJPE in dark environments (D), EPM increases by +37.7%, whereas VIBE significantly jumps to +72.0%.

This disparity suggests that VIBE, when estimating meshes in dark environments, often misestimates the orientation of the person, confusing the front and back. The top row of Fig. 7 presents the estimation results of EPM and VIBE in light environments, while the bottom row shows the results in dark environments. It is evident in the dark environments that VIBE reverses the orientation of the person compared to the ground truth mesh. This phenomenon leads to a relatively larger error in PEL-MPJPE, which calculates error after aligning the root joint's position only, compared to PA-MPJPE, which aligns position, rotation, and scale for error calculation. The results from both quantitative and qualitative analyses indicate that VIBE, which utilizes traditional frame-based cameras, experiences a significant degradation in estimation accuracy in dark environments. In contrast, EventPointMesh, which solely relies on event data, shows a minor decrease in estimation accuracy. These findings demonstrate the high robustness of our proposed method against dim lighting conditions.

D. Ablation Study

We conduct an ablation study to verify the contribution of our coarse-to-fine feature extraction to the accuracy of mesh

TABLE IV
QUANTITATIVE RESULTS FROM THE ABLATION STUDY

Method	MPJPE [mm] (↓)	PEL-MPJPE [mm] (↓)	PA-MPJPE [mm] (↓)	PVE [mm] (↓)	PCKh@0.5 [%] (↑)
EPM(G)	115.8	89.3	63.2	73.3	69.9
EPM(L)	116.0	87.0	61.8	71.9	71.0
EPM(G+L)	**110.7**	**86.0**	**60.7**	**71.8**	**71.7**

TABLE V
QUANTITATIVE RESULTS FROM THE PERFORMANCE EVALUATION UNDER VARIOUS INPUT CONDITIONS

	Method	MPJPE [mm] (↓)	PEL-MPJPE [mm] (↓)	PA-MPJPE [mm] (↓)	PVE [mm] (↓)	PCKh@0.5 [%] (↑)
Task1	EPM(1)	167.3	138.1	97.2	111.0	51.7
	EPM(5)	125.2	100.8	70.5	82.4	65.8
	EPM(15)	**110.7**	**86.0**	**60.7**	**71.8**	**71.7**
Task2	EPM(120)	160.6	129.3	90.6	105.6	54.8
	EPM(img)	**110.7**	**86.0**	**60.7**	**71.8**	**71.7**

estimation. The architecture of our EPM network leverages the global features from the Keypoints Module and the local features from the Anchor Points Module. Both these features are integrated within the SMPL Module. To illustrate the contributions of these elements to mesh estimation, we evaluate the accuracy of estimation when each component is omitted. We denote the method that estimates meshes using only global features as EPM(G), the method using only local features as EPM(L), and the method leveraging both as EPM(G+L). The results of the experiments are presented in Table IV.

The results indicated that EPM(G+L) achieved the highest scores across all metrics. When comparing EPM(G) and EPM(L), it was found that EPM(G) had higher accuracy in MPJPE, while EPM(L) scored higher in both PEL-MPJPE and PA-MPJPE. MPJPE is a metric that evaluates the 3D euclidean distance of joint coordinates without aligning the estimated and ground true poses in terms of position, rotation, and scale. Therefore, it is reasonable that EPM(G), which captures the coarse global features of human pose, 3D position, and body shape, performed better in MPJPE. On the other hand, PEL-MPJPE applies MPJPE after aligning the position of the root joint, and PA-MPJPE applies MPJPE after aligning translation, rotation, and scale. The higher accuracy of EPM(L) in these metrics suggests that focusing on the finer details around each joint led to better precision than EPM(G). As global and local features can be complementary, the combined use of both in EPM(G+L) resulted in further improved accuracy over either approach alone. Therefore, our pioneering coarse-to-fine feature extraction approach, initially extracting coarse global features of the event point cloud and then grouping the spatiotemporal event point cloud around each joint to extract detailed local features, demonstrates effectiveness in HMR using only event data.

E. Performance Evaluation Under Various Input Conditions

Task1: Different Length of Input Event Sequence. This experiment was conducted to evaluate the effectiveness of feeding the event point cloud into the network as sequential temporal data. In our EPM network, we incorporate BiLSTM into both the Keypoints Module and Anchor Points Modules to consider the temporal characteristics of the input event point clouds. We demonstrate the efficacy of the proposed network architecture that accounts for temporal features by evaluating the accuracy of the estimated mesh from varying lengths of event data. In this study, the event point clouds fed into the network were divided into individual time windows between intensity image frames. We experimented with three patterns: feeding event point clouds from 1, 5, and 15 continuous time windows into the model, denoted as EPM(1), EPM(5), and EPM(15) respectively. The experimental results are shown in Table V.

The results indicate that EPM(15) yielded the highest mesh estimation accuracy, demonstrating the effectiveness of extracting temporal features for mesh estimation. Due to the event cameras' nature, they don't capture static parts of the subject. This makes estimating the entire body mesh from brief event data segments, such as a single time window, challenging. However, our proposed network overcomes these limitations, enabling high-precision mesh estimation.

Task2: Different Width of Time-Window: We evaluate the precision of mesh estimation over different time-window widths, assessing how adaptably the EventPointMesh network can handle variations in time-window sizes. In previously discussed experiments, event data were segmented according to the frame rate of intensity images captured by the event camera. The frame rate of intensity images within the EPMD is roughly 15 fps under well-lit conditions and about 4 fps in poorly-lit scenarios. Prior research, EventHPE [13], estimated the mesh synchronized to the frame rate of the intensity images taken with the event camera. This approach inherently limits the maximum frame rate of the output mesh to that of the intensity images. In contrast, our EventPointMesh does not rely on intensity images, allowing it to process input event data at any chosen time window. Consequently, we carried out training and testing with event data segmented to match the 120 fps time window, corresponding to the optical MoCap [1] data.

The experimental results are presented in Fig. 8 and Table V. The method segmented based on the frame rate of intensity images is labeled as EPM(img), while the one conducted at

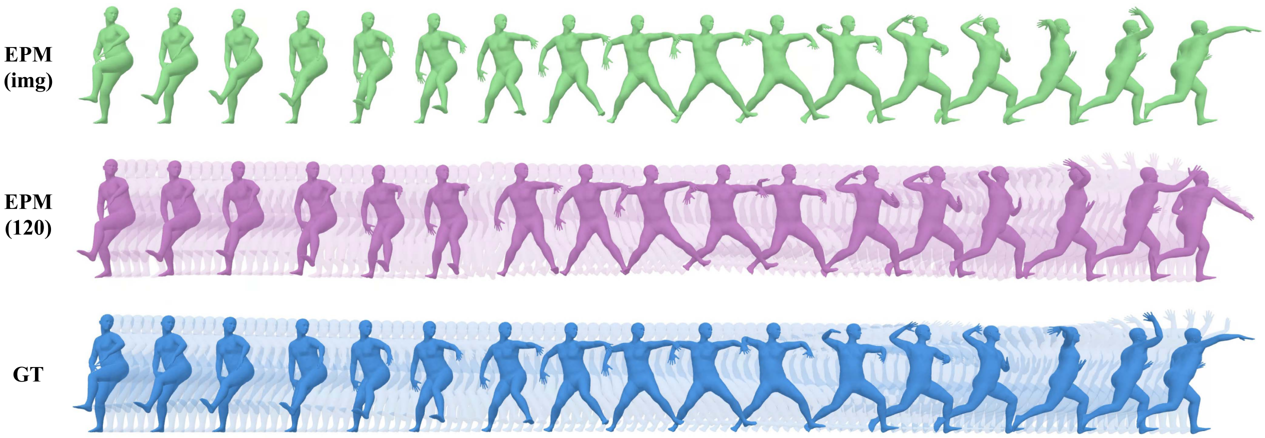

Fig. 8. Qualitative results of the Task2 described in Section IV-E. The topmost "EPM(img)" represents our method's results when segmenting events according to the intensity image frame rate, which varies with the lighting conditions. The middle "EPM(120)" shows the outcomes when segmenting events and estimating the meshes at the same 120 fps. The bottom "GT" shows the ground truth meshes with a frame rate of 120 fps. Note that the transparent meshes in the bottom two rows are only for visual clarity; there is no difference from the opaque meshes.

120 fps is defined as EPM(120). The results show that although EPM(120) is slightly less effective than EPM(img), it still possesses sufficient capability for mesh estimation. The decrease in accuracy for EPM(120) compared to EPM(img) can be attributed to the significantly reduced amount of event information in the time window when operating at a high frame rate and the subject's movements are slow. In this experiment, the number of events per time window is set to 7500, and if the detected number of events falls below this, points are randomly duplicated to increase to 7500, but this does not add useful information for reconstructing the movement. Therefore, at this stage, increasing the frame rate reduces the number of events per time window, thereby decreasing the estimation accuracy. Addressing this issue by developing a mechanism that can extract useful information from a small number of events, rather than randomly duplicating them, could lead to the construction of a higher frame rate method.

Fig. 8 demonstrates that our proposed method can achieve HMR at a high frame rate comparable to traditional optical motion capture (MoCap). This indicates that even in challenging environments for traditional frame-based cameras, such as poorly-lit conditions like dark rooms and high-speed subject movements, our method enables a simplified 3D HMR with just a single monocular event camera.

V. Discussion and Limitations

A. Failure Cases

Our method has liberated event-based HMR from dependence on intensity images, enhancing its applicability in real-world scenarios. Moreover, although our approach involves bidirectionality in its processing, it can sequentially estimate meshes from event point clouds, making it feasible for transition to online applications. Specifically, there is potential for utilizing our method in activity monitoring systems within environments requiring privacy, such as hospitals, care facilities, and homes. Additionally, for existing efforts in developing projection mapping that projects video fitted to the body of performers on stage in real-time, our method offers a highly beneficial system capable of handling fast movements in dark environments. However, towards the realization of such applications, there are several scenarios where we currently face challenges, necessitating improvements. Below, we discuss several representative failure cases and the corresponding strategies for addressing them.

Case 1: Few or no Detected Events. This case considers three types of scenarios. The first and most common scenario is when there is minimal movement of the body. Event cameras function by detecting per-pixel changes in luminance. Thus, movements that are not significant but still result in changes in luminance values will be detected as events. Since EventPointMesh utilizes time-series point cloud data as input, it is capable of estimating the most plausible pose for the entire body by leveraging the features from event point clouds of other moving body parts, even if events for a specific part of the body are temporarily sparse. This capability is evidenced in Fig. 5, where accurate mesh estimations are achieved even in instances where a body part is not captured in the events.

However, with the current network architecture, if the 2D joint positions estimated by the Keypoint Module are incorrect, it can result in the inability to extract useful information from the point cloud for mesh estimation in that local area, or it might capture features that adversely affect the estimations of subsequent modules. One effective solution to this problem could involve modifying the loss function used in the Keypoint Module for estimating 2D joint positions from KL divergence to Negative Log-Likelihood. This adjustment would allow the estimation process to also yield confidence levels, which could then be used by the Anchor Points Module to more accurately extract local features. Consequently, this could enhance the transmission of only the most beneficial local features to subsequent modules. The potential benefits of this refinement will be explored in future research to develop the method further.

Nevertheless, the current system, which utilizes BiLSTM to account for temporal relationships, can result in the output of meshes with high uncertainty in situations where there is a prolonged static state, as it may lead to the loss of

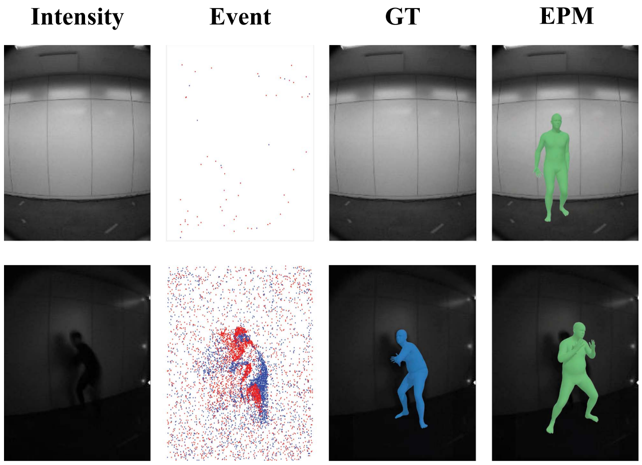

Fig. 9. Failure cases of mesh estimation by EventPointMesh. The top row presents the output of our method for a scenario discussed in V-A Case 1, where the subject steps out of the camera's view. Currently, in cases where no events are generated due to the absence of the subject, a mesh in a typical standing pose is outputted. The bottom row shows an example of mesh estimation when events are generated by parts other than the subject, as discussed in V-A Case 2. It demonstrates that the presence of shadows causing events can lead to a reduction in mesh estimation accuracy.

information supporting the presence of certain body parts. An effective solution to such long-term event omissions could be the implementation of autoregressive network architectures. An autoregressive network incorporates a loop structure that feeds the pose information, the output of the current time window, combined with the event data for the next time window as input to estimate the pose for the next time window. This means that once the pose has been estimated from the events associated with movement of body parts, the existence of these parts can still be inferred in their last known positions even if no new events are generated due to their lack of movement. Hence, it is anticipated that accurate estimations of the entire body pose at any given moment can be made based on the pose information carried over from previous time windows, even in cases where no events are detected over an extended period.

The second scenario involves situations where parts or the entirety of the subject's body move out of the camera's field of view, a common occurrence in real-world applications utilizing video-based pose estimation methods. The EPMD dataset assumes that the subject remains within the camera's field of view, with almost all data capturing the full body. Consequently, the pose estimation accuracy decreases in this scenario due to the network being trained on such data. When only a small part of the body (such as the hands, feet, or head) moves out of view, the system can still estimate a plausible 3D body model that includes the body parts located outside the 2D image frame by utilizing temporal point cloud features and the structural constraints of the parametric human model. However, as the amount of the body outside the field of view increases, the accuracy of position and pose estimation deteriorates, ultimately leading to an unexpected output as illustrated in the upper part of Fig. 9. Ideally, no mesh model should be generated when the subject is outside the camera's view, as no events occur within the frame. However, the current system produces a standing pose mesh, likely because the model outputs the most common pose within the dataset when faced with such an unfamiliar scenario. To address this issue, incorporating a mechanism commonly known as the "top-down approach" in RGB image-based pose estimation methods, which estimates poses based on bounding boxes detected as human regions, could effectively handle situations where the subject moves in and out of the field of view.

The third scenario occurs when insufficient lighting prevent the generation of events. The EPMD dataset includes event data captured under both well-lit and poorly-lit lighting conditions. In Fig. 6, event data captured in dim environments appears to have more background noise compared to that from well-lit environments at first glance. However, this appearance is due to accumulating event point clouds over long time windows to match the frame rate of intensity images, creating a single event frame. Thus, event cameras fundamentally are not adversely affected by dark environments and can accurately capture the movements of subjects with high temporal resolution without blurring. Nonetheless, poor lighting can impact the accuracy of event data. This issue arises in lighting conditions so dim that the changes in luminance caused by the subject's movement do not exceed the threshold set on the event camera device for detecting event occurrences. In such extremely dark environments, it might be preferable to use night vision cameras utilizing infrared instead of event cameras.

Case 2: Events Caused by Entities Other Than the Subject. Our method faces challenges even when events are generated by entities other than a single subject, such as multiple people within the frame or movements of the camera itself. Our EPMD dataset is specifically designed for a stationary event camera capturing a single subject. When other humans or dynamic objects enter the field of view, or the camera moves, events unrelated to the subject's movement are detected, which can significantly compromise the accuracy of mesh estimation. Indeed, some data in the EPMD dataset captured in dim environments show shadows cast on the wall behind the subject. The bottom row of Fig. 9 presents an example where such conditions have led to a decrease in estimation accuracy. As illustrated by the intensity and event images in the figure, the presence of a shadow on the left side of the subject can disrupt event patterns to an extent that currently impacts mesh estimation accuracy. Therefore, the addition of other people's movements or camera motion is expected to further reduce estimation accuracy. One potential solution to this problem is to enrich the dataset with real or synthesized data capturing scenes with multiple humans or objects, or where the camera is in motion. Alternatively, introducing detectors and trackers to segment event data by individual or object and estimating the intended subject's pose could be an effective approach.

B. Room for Improvement in Event-Based HMR

Existing event-based HMR methods, including this study, still leave an important aspect unaddressed: the ability to accurately represent the clothing worn by the subjects. As demonstrated by the results of the experiment in the Section IV-B "Evaluation on MMHPSD (Cross-Subject)", differences in clothing and individual subject movements lead to variations in events, which in turn affect the accuracy of mesh reconstruction. This issue becomes more pronounced in EPMD, where the limited number of subjects worsens the problem. Addressing this challenge necessitates the creation of more diverse datasets through the augmentation of subject numbers and the use of data synthesis with computer graphics in future work.

In addition to this issue, there are more fundamental problems that require deeper solutions. Both the MMHPSD and EPMD datasets utilize the SMPL model to replicate the body shape including clothing. However, models trained on such datasets face limitations in generalizing across diverse clothing types when applied in real-world settings. While MMHPSD attempts to fit the SMPL model to the 3D shapes reconstructed using multiple RGB-D cameras around subjects in various outfits, this approach struggles with accurately representing oversized clothes or skirts. Similarly, EPMD employs optical MoCap to capture motion data, necessitating the use of specialized suits for attaching optical markers. Consequently, the placement and number of markers can alter the shape of the SMPL model, also presenting challenges in accommodating diverse clothing styles.

While this issue is also present in HMR using traditional frame-based cameras, it is particularly pressing for event cameras due to their principle of capturing only the subject's movement. The fluctuation of clothing has a significant impact on the captured event data, demanding more thorough solutions. In future research, one potential solution could involve constructing mesh models that differentiate between the subject's body shape and the shape of their clothing. Additionally, developing mechanisms capable of distinguishing whether captured events are caused by body movement or clothing movement will also be crucial for achieving more accurate HMR.

VI. Conclusion

This paper explored the potential of using event data exclusively for effective human mesh recovery (HMR), bypassing the limitations of conventional intensity images. Our proposed EventPointMesh framework demonstrated superior performance by interpreting event data as a 3D spatio-temporal point cloud and leveraging both global and local features. The series of experiments and the ablation study affirmed the efficacy of our approach, especially under varied lighting conditions. However, several challenges remain, such as capturing static body parts in event data, people entering and exiting the frame, and events caused by objects other than the subject. Future endeavors will focus on enhancing accuracy and adaptability in real-world scenarios, but our current achievements represent a significant leap in event-based HMR, paving the way for efficient methodologies in challenging conditions.

References

[1] OptiTrack, "Optitrack motion capture systems," 2009. [Online]. Available: https://www.optitrack.com/

[2] Vicon, "Vicon motion capture systems," 2010. [Online]. Available: https://www.vicon.com/

[3] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.

[4] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional multi-person pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2353–2362.

[5] J. Wang et al., "Deep 3D human pose estimation: A review," *Comput. Vis. Image Understanding*, vol. 210, 2021, Art. no. 103225.

[6] A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik, "Learning 3D human dynamics from video," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5607–5616.

[7] M. Kocabas, N. Athanasiou, and M. J. Black, "VIBE: Video inference for human body pose and shape estimation," *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5252–5262.

[8] Y. Tian, H. Zhang, Y. Liu, and L. Wang, "Recovering 3D human mesh from monocular images: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 15406–15425, Dec. 2023.

[9] M. Zhao et al., "Through-wall human mesh recovery using radio signals," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 10 112–10 121.

[10] H. Xue et al., "MmMesh: Towards 3D real-time dynamic human mesh construction using millimeter-wave," in *Proc. ACM Int. Conf. Mobile Syst., Appl., Serv.*, 2021, pp. 269–282.

[11] G. Gallego et al., "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, Jan. 2022.

[12] L. Xu, W. Xu, V. Golyanik, M. Habermann, L. Fang, and C. Theobalt, "EventCap: Monocular 3D capture of high-speed human motions using an event camera," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4967–4977.

[13] S. Zou et al., "EventHPE: Event-based 3D human pose and shape estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10 976–10 985.

[14] S. Zou, Y. Mu, X. Zuo, S. Wang, and L. Cheng, "Event-based human pose tracking by spiking spatiotemporal transformer," 2023, *arXiv:2303.09681*.

[15] C. Bregler and J. Malik, "Tracking people with twists and exponential maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1998, pp. 8–15.

[16] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun, "Performance capture from sparse multi-view video," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–10, 2008.

[17] L. Sigal, A. Balan, and M. J. Black, "HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *Int. J. Comput. Vis.*, vol. 87, no. 1, pp. 4–27, 2010.

[18] L. Sigal, M. Isard, H. Haussecker, and M. J. Black, "Loose-limbed People: Estimating 3D human pose and motion using non-parametric belief propagation," *Int. J. Comput. Vis.*, vol. 98, no. 1, pp. 15–48, 2011.

[19] C. Stoll, N. Hasler, J. Gall, H.-P. Seidel, and C. Theobalt, "Fast articulated motion tracking using a sums of gaussians body model," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 951–958.

[20] M. B. Holte, C. Tran, M. M. Trivedi, and T. B. Moeslund, "Human pose estimation and activity recognition from multi-view videos: Comparative explorations of recent developments," *IEEE J. Sel. Top. Signal Process.*, vol. 6, no. 5, pp. 538–552, Sep. 2012.

[21] M. Burenius, J. Sullivan, and S. Carlsson, "3D pictorial structures for multiple view articulated pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3618–3625.

[22] H. Joo et al., "Panoptic Studio: A massively multiview system for social motion capture," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2015, pp. 3334–3342.

[23] A. Elhayek et al., "Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3810–3818.

[24] H. Rhodin, N. Robertini, C. Richardt, H.-P. Seidel, and C. Theobalt, "A versatile scene model with differentiable visibility applied to generative pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 765–773.

[25] N. Robertini, D. Casas, H. Rhodin, H.-P. Seidel, and C. Theobalt, "Model-based outdoor performance capture," in *Proc. Int. Conf. 3D Vis.*, 2016, pp. 166–175.

[26] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Harvesting multiple views for marker-less 3D human pose annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1253–1262.

[27] L. Xu et al., "FlyCap: Markerless motion capture using multiple autonomous flying cameras," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 8, pp. 2284–2297, Aug. 2018.

[28] S. Li and A. B. Chan, "3D human pose estimation from monocular images with deep convolutional neural network," in *Proc. Asian Conf. Comput. Vis.*, 2015, pp. 332–347.

[29] D. Mehta et al., "Monocular 3D human pose estimation in the wild using improved cnn supervision," in *Proc. Int. Conf. 3D Vis.*, 2017, pp. 506–516.

[30] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3D human pose," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1263–1272.

[31] D. Mehta et al., "VNect: Real-time 3D human pose estimation with a single RGB camera," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–14, 2017.

[32] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis, "Sparseness meets deepness: 3D human pose estimation from monocular video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4966–4975.

[33] C. Chen and D. Ramanan, "3D human pose estimation = 2D pose estimation + matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5759–5767.

[34] H. Yasin, U. Iqbal, B. Krüger, A. Weber, and J. Gall, "A dual-source approach for 3D pose estimation from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4948–4956.

[35] E. Jahangiri and A. L. Yuille, "Generating multiple diverse hypotheses for human 3D pose consistent with 2D joint detections," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, 2017, pp. 805–814.

[36] Y. Chen, Y. Tian, and M. He, "Monocular human pose estimation: A survey of deep learning-based methods," *Comput. Vis. Image Understanding*, vol. 192, 2020, Art. no. 102897.

[37] L. Mourot, L. Hoyet, F. Le Clerc, F. Schnitzler, and P. Hellier, "A survey on deep learning for skeleton-based human animation," *Comput. Graph. Forum*, vol. 41, no. 1, pp. 122–157, 2022.

[38] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–16, 2015.

[39] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep It SMPL: Automatic estimation of 3D human pose and shape from a single image," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 561–578.

[40] N. Kolotouros, G. Pavlakos, and K. Daniilidis, "Convolutional mesh regression for single-image human shape reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4496–4505.

[41] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler, "Unite the people: Closing the loop between 3D and 2D human representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4704–4713.

[42] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7122–7131.

[43] A. Kowdle et al., "The need 4 speed in real-time dense visual tracking," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–14, 2018.

[44] M.-Z. Yuan, L. Gao, H. Fu, and S. Xia, "Temporal upsampling of depth maps using a hybrid camera," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 3, pp. 1591–1602, Mar. 2019.

[45] Y. Ren, Z. Wang, Y. Wang, S. Tan, Y. Chen, and J. Yang, "GoPose: 3D human pose estimation using WiFi," *ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 2, pp. 1–25, 2022.

[46] W. Jiang et al., "Towards 3D human pose construction using WiFi," in *Proc. Int. Conf. Mob. Comput. Netw.*, 2020, pp. 1–14.

[47] M. Zhao et al., "RF-based 3D skeletons," in *Proc. Conf. ACM Spec. Int. Group Data Commun.*, 2018, pp. 267–281.

[48] T. Li, L. Fan, M. Zhao, Y. Liu, and D. Katabi, "Making the invisible visible: Action recognition through walls and occlusions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 872–881.

[49] A. Sengupta, F. Jin, R. Zhang, and S. Cao, "mm-Pose: Real-time human skeletal posture estimation using mmWave radars and CNNs," *IEEE Sensors J.*, vol. 20, no. 17, pp. 10 032–10 044, Sep. 2020.

[50] S. An and U. Y. Ogras, "Fast and scalable human pose estimation using mmwave point cloud," in *Proc. ACM/IEEE Des. Autom. Conf.*, 2022, pp. 889–894.

[51] G. Chen et al., "Neuromorphic vision-based fall localization in event streams with temporal–spatial attention weighted network," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9251–9262, Sep. 2022.

[52] L. Sun et al., "Event-based fusion for motion deblurring with cross-modal attention," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 412–428.

[53] I. Alonso and A. C. Murillo, "EV-SegNet: Semantic segmentation for event-based cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 1624–1633.

[54] J. Zhang, K. Yang, and R. Stiefelhagen, "ISSAFE: Improving semantic segmentation in accidents by fusing event-based data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 1132–1139.

[55] Y. Zhou, G. Gallego, and S. Shen, "Event-based stereo visual odometry," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1433–1450, Oct. 2021.

[56] I. Alzugaray and M. Chli, "Asynchronous multi-hypothesis tracking of features with event cameras," in *Proc. Int. Conf. 3D Vis.*, 2019, pp. 269–278.

[57] Y. Li et al., "Graph-based asynchronous event processing for rapid object recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 914–923.

[58] Y. Wang et al., "EV-Gait: Event-based robust gait recognition using dynamic vision sensors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6351–6360.

[59] G. Gallego, H. Rebecq, and D. Scaramuzza, "A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3867–3876.

[60] L. Pan, M. Liu, and R. Hartley, "Single image optical flow estimation with an event camera," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1669–1678.

[61] J. Hidalgo-Carrió, D. Gehrig, and D. Scaramuzza, "Learning monocular dense depth from events," in *Proc. Int. Conf. 3D Vis.*, 2020, pp. 534–542.

[62] J. Jiao, H. Huang, L. Li, Z. He, Y. Zhu, and M. Liu, "Comparing representations in tracking for event camera-based SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2021, pp. 1369–1376.

[63] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll, "Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception," *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 34–49, Jul. 2020.

[64] E. Calabrese et al., "DHP19: Dynamic vision sensor 3D human pose dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 1695–1704.

[65] G. Scarpellini, P. Morerio, and A. Del Bue, "Lifting monocular events to 3D human poses," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2021, pp. 1358–1368.

[66] Z. Zhang et al., "Neuromorphic high-frequency 3D dancing pose estimation in dynamic environment," *Neurocomputing*, vol. 547, 2023, Art. no. 126388.

[67] J. Chen, H. Shi, Y. Ye, K. Yang, L. Sun, and K. Wang, "Efficient human pose estimation via 3D event point cloud," in *Proc. Int. Conf. 3D Vis.*, 2022, pp. 1–10.

[68] Z. Shao, X. Wang, W. Zhou, W. Wang, J. Yang, and Y. Li, "A temporal densely connected recurrent network for event-based human pose estimation," *Pattern Recognit.*, vol. 147, 2023, Art. no. 110048.

[69] R. W. Baldwin, R. Liu, M. Almatrafi, V. Asari, and K. Hirakawa, "Time-ordered recent event (TORE) volumes for event cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2519–2532, Feb. 2023.

[70] G. Goyal, F. Di Pietro, N. Carissimi, A. Glover, and C. Bartolozzi, "MoveEnet: Online high-frequency human pose estimation with an event camera," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2023, pp. 4024–4033.

[71] L. Gava, M. Monforte, C. Bartolozzi, and A. Glover, "How late is too late? A preliminary event-based latency evaluation," in *Proc. Int. Conf. Event-Based Control Commun. Signal Process.*, 2022, pp. 1–4.

[72] V. Rudnev et al., "EventHands: Real-time neural 3D hand pose estimation from an event stream," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 12 365–12 375.

[73] J. Li et al., "LiDARCap: Long-range marker-less 3D human motion capture with lidar point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 20 502–20 512.

[74] Y. Ren et al., "LiDAR-aid inertial poser: Large-scale human motion capture by sparse inertial and LiDAR sensors," *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 5, pp. 2337–2347, May 2023.

[75] Y. Zhou, H. Dong, and A. E. Saddik, "Learning to estimate 3D human pose from point cloud," *IEEE Sensors J.*, vol. 20, no. 20, pp. 12 334–12 342, Oct. 2020.

[76] S. Wang, A. Geiger, and S. Tang, "Locally aware piecewise transformation fields for 3D human mesh registration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7635–7644.

[77] X. Zuo, S. Wang, Q. Sun, M. Gong, and L. Cheng, "Self-supervised 3D human mesh recovery from noisy point clouds," 2021, *arXiv:2107.07539*.

[78] H. Feng, P. Kulits, S. Liu, M. J. Black, and V. F. Abrevaya, "Generalizing neural human fitting to unseen poses with articulated Se(3) equivariance," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 7977–7988.

[79] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.

[80] Y. Li et al., “SimCC: A simple coordinate classification perspective for human pose estimation,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 89–106.

[81] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, “A 240 × 180 130 DB 3 µs latency global shutter spatiotemporal vision sensor,” *IEEE J. Solid-Statist. Circuits*, vol. 49, no. 10, pp. 2333–2341, Oct. 2014.

[82] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. Black, “AMASS: Archive of motion capture as surface shapes,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5441–5450.

[83] M. M. Loper, N. Mahmood, and M. J. Black, “MoSh: Motion and shape capture from sparse markers,” in *Proc. SIGGRAPH Asia*, vol. 33, no. 6, pp. 220:1–220:13, 2014.

[84] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Representations*, 2015.

[85] J. Gower, “Generalized procrustes analysis,” *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975. [Online]. Available: https://EconPapers.repec.org/RePEc:spr:psycho:v:40:y:1975:i:1:p:33--51

Ryosuke Hori (Student Member, IEEE) received the BE and MScEng degree in information and computer science from Keio University, Japan, in 2021 and 2022 respectively. He is currently working toward the PhD degree in science and technology at Keio University. His research interests include 3D human pose and shape estimation and neuromorphic vision.

Mariko Isogawa (Member, IEEE) received the BS, MS, and PhD degrees from Osaka University, Japan, in 2011, 2013, and 2019, respectively. From 2019 to 2020, she was a visiting scholar with Carnegie Mellon University, USA. She is currently an associate professor with the Department of Information and Computer Science, Faculty of Science and Technology, Keio University, Japan. Her research interests include computer vision and pattern recognition.

Dan Mikami (Member, IEEE) received the BE and ME degrees from Keio University, Japan, in 2000 and 2002, respectively, and the PhD degree from the University of Tsukuba, Japan, in 2012. In 2002, he joined the Nippon Telegraph and Telephone Corporation (NTT). Since 2021, he has held the position of associate professor with the Faculty of Informatics, Kogakuin University. His research interests encompass computer vision, virtual reality, and computer-aided motor learning.

Hideo Saito (Senior Member, IEEE) received the PhD degree in electrical engineering from Keio University, Japan, in 1992. Since 1992, he has been in the Faculty of Science and Technology, Keio University. From 1997 to 1999, he joined the Virtualized Reality Project with the Robotics Institute, Carnegie Mellon University, as a visiting researcher. Since 2006, he has been a full professor with the Department of Information and Computer Science, Keio University. His research interests include computer vision and pattern recognition, and their applications to augmented reality, virtual reality, and human–robotic interaction. His recent activities in academic conferences include being the Program Chair of ACCV 2014, the General Chair of ISMAR 2015, and the Program Chair of ISMAR 2016.