DHP19: Dynamic Vision Sensor 3D Human Pose Dataset

Enrico Calabrese[†,*], Gemma Taverni[†,*], Christopher Awai Easthope[‡], Sophie Skriabine[†], Federico Corradi[†], Luca Longinotti[≀], Kynan Eng[≀,†], Tobi Delbruck[†]

[†]Institute of Neuroinformatics, University of Zurich and ETH Zurich,
[‡]Balgrist University Hospital, University of Zurich, [≀]iniVation AG, Zurich

Abstract

Human pose estimation has dramatically improved thanks to the continuous developments in deep learning. However, marker-free human pose estimation based on standard frame-based cameras is still slow and power hungry for real-time feedback interaction because of the huge number of operations necessary for large Convolutional Neural Network (CNN) inference. Event-based cameras such as the Dynamic Vision Sensor (DVS) quickly output sparse moving-edge information. Their sparse and rapid output is ideal for driving low-latency CNNs, thus potentially allowing real-time interaction for human pose estimators. Although the application of CNNs to standard frame-based cameras for human pose estimation is well established, their application to event-based cameras is still under study. This paper proposes a novel benchmark dataset of human body movements, the Dynamic Vision Sensor Human Pose dataset (DHP19). It consists of recordings from 4 synchronized 346x260 pixel DVS cameras, for a set of 33 movements with 17 subjects. DHP19 also includes a 3D pose estimation model that achieves an average 3D pose estimation error of about 8 cm, despite the sparse and reduced input data from the DVS.

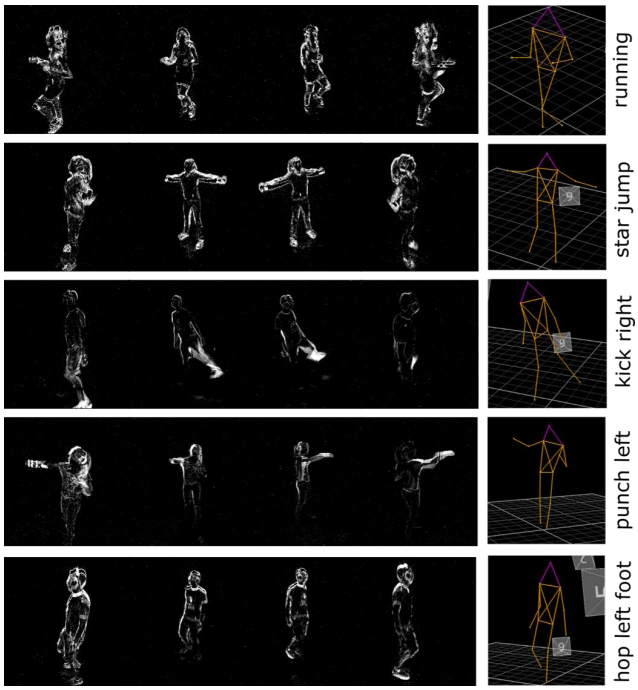

Figure 1. Examples from DHP19: DVS recordings (left) and Vicon labels (right) from 5 of the 33 movements. For visualization, the DVS events are here accumulated into frames (about 7.5 k events per single camera), following the procedure described in Sec. 4.

DHP19 Dataset

DHP19 dataset and code are available at:
https://sites.google.com/view/dhp19.

1. Introduction

Conventional video technology is based on a sequence of static frames captured at a fixed frame rate. This comes with several drawbacks, such as: large parts of the data are redundant, the background information is recorded at every frame, and the information related to the moving objects is limited by the frame rate of the camera. Recently, event cameras have proposed a paradigm shift in vision sensor technology, providing a continuous and asynchronous stream of brightness-change events. Event cameras, such as the Dynamic Vision Sensor (**DVS**) [7, 17], grant higher dynamic range and higher temporal resolution at a lower power budget and reduced data-transfer bandwidth when compared to conventional frame-based cameras [17]. The redundancy reduction and high sparsity provided by the

*E-mail: {enrico, getaverni}@ini.uzh.ch (equal contribution)

DVS camera can make processing algorithms both memory and computationally lighter, while preserving the significant information to be processed. Indeed, the properties of the DVS camera have made it an attractive candidate for applications in motion-related tasks [12, 20]. Moreover, previous work [21] has demonstrated that the DVS sparse representation and high dynamic range can facilitate learning in Convolutional Neural Networks (**CNNs**) compared to standard frame-based input. Until now, CNNs applied to the output of event cameras have been proposed to solve classification [5, 19, 20] and single output regression tasks [21], but this has (to our knowledge) never been attempted for multiple output regression problems.

In this paper, we introduce the first DVS benchmark dataset for multi-view 3D human pose estimation (**HPE**), where the goal is to recover the 3D position of human joints visible in event streams recorded from multiple DVS cameras. In particular, we aim at exploring the application of DVS cameras in combination with new HPE techniques for more efficient online processing. In fact, HPE has broad application in the real-time domain, where low-latency pose prediction is an important attribute, such as virtual reality, gaming, accident detection, and real-time movement feedback in rehabilitation therapy. State-of-the-art techniques have experimented the use of frame-based cameras in combination with CNNs reaching high level of accuracy. Although CNNs represent the leading method in HPE, and more generally in the whole visual recognition field, current solutions still suffer from drawbacks in terms of large GPU requirements and long learning phases. Those features make them too slow or too power hungry for some real-time applications. Therefore, there is a growing need for efficient HPE, while retaining robustness and accuracy. For these reasons, in this paper we explore the application of DVS event-based cameras for HPE.

The main contributions of this paper are: we introduce the Dynamic Vision Sensor Human Pose dataset (**DHP19**), the first DVS dataset for 3D human pose estimation. DHP19 includes synchronized recordings from 4 DVS cameras of 33 different movements (each repeated 10 times) from 17 subjects, and the 3D position of 13 joints acquired with the Vicon motion capture system [2]. Furthermore, a reference study is presented performing 3D HPE on DHP19. In particular, we train a CNN on multi-camera input for 2D HPE, and use geometric information for 3D reconstruction using triangulation. Our proposed approach achieves an average joint position error comparable to state-of-the-art models.

2. Related work

2.1. DVS and DAVIS sensors

The DVS camera responds to changes in brightness. Each pixel works independently and asynchronously. The pixel generates a new event when the logarithm of the incoming light changes by a specific threshold from the last event, and the new brightness is memorized. In a static-camera setup, the data generated by the DVS camera contains only information about moving objects and the background is automatically subtracted at the sensor stage. The camera output is a stream of events, each represented by the time it occurred (in microseconds), the (x,y) address of the pixel, and the sign of the brightness change [7, 17]. Moreover, the logarithmic response provides an intrascene dynamic range of over 100 dB, which is ideal for applications under the wide range of natural lighting conditions. Fig. 2a) shows the DVS working principle. Events are generated in a wide dynamic range, responding to contrast changes. The event cameras used in this paper are of the Dynamic and Active Pixel Vision Sensor (**DAVIS**) type, an advanced version of the DVS [17]. The DAVIS camera is able to record both DVS events and standard static APS (Active Pixel Sensor) frames. Fig. 2b) shows the difference between the APS frame and the DVS stream of events.

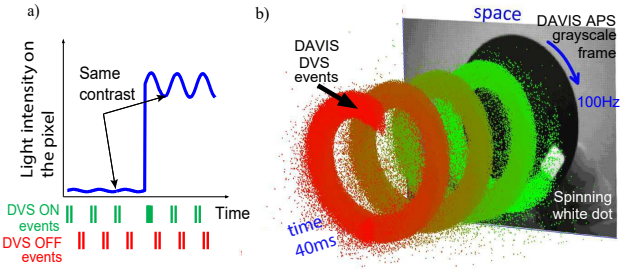

Figure 2. a) A DVS pixel generates log intensity change events, representing local reflectance changes working in a wide range of light condition. b) DAVIS grayscale frame and events generated from a spinning dot; the sparse event output shows the rapidly moving dot, otherwise blurred in the grayscale frame.

Table 1. Frame- (F) and event-based (E) datasets for 3D HPE.

Name	Type	# Cam.	# Subj.	# Mov.	Eval. Metric
HumanEva [29]	F	3/4	4	6	MPJPE[1]
Human3.6M [16]	F	4	11	15	MPJPE[1]
MPI-INF-3DHP [22]	F	14	8	8	MPJPE[1], PCK[2]
MADS [32]	F	3	5	30	MPJPE[1]
DHP19 (This work)	E	4	17	33	MPJPE[1]

[1] Mean per joint position error (mm), [2] Percentage of correct keypoints (%)

2.2. Event-based datasets

To date there are only a limited number of published event-based datasets, due to the relative novelty of the technology [1]. Among these, only two relate to human gestures or body movements. [14] includes DVS data for action recognition from the VOT2015 and UFC50 datasets converted from standard video to DVS data by displaying the frame-based dataset on a 60 Hz LCD monitor in front a DVS camera (DAVIS240 [7]). [5] introduced a dataset of 11 hand gestures from 29 subjects, for gesture classification. A

DVS128 [17] was used to record the upper-body part of the subject performing the actions. In this case, the spatial resolution of the DVS128 is relatively low (128x128 pixel) and the variety of movements is restricted to hand actions. No existing DVS dataset includes joint positions.

2.3. Human pose datasets

Existing datasets for 3D HPE are recorded using frame-based cameras, and the large majority include RGB color channels recordings. The most commonly used datasets are: HumanEva [29], Human3.6M [16], MPI-INF-3DHP [22] and MADS [32]. All of these datasets include multi-view camera recordings of the whole body of subjects performing different movements, and include ground truth 3D pose recording from a motion capture system. The datasets are recorded in a lab environment. In addition, MPI-INF-3DHP is recorded using a green screen background for automatic segmentation and allows for wild background addition. Table 1 highlights the main characteristics of the existing RGB frame-based 3D HPE datasets and our DHP19 event-based dataset.

2.4. CNNs for 3D human pose estimation

In recent years CNNs have emerged as the most successful method for computer vision recognition, including 3D HPE. For 3D HPE, existing approaches reconstruct the 3D pose from single [9, 23, 24, 27, 30, 33] or multiple [4, 11, 28] camera views. Multi-view methods are superior to single-view in that they reduce occlusion and can solve ambiguities, increasing prediction accuracy and robustness. However, they require a more complex setup, increase the amount of input information, and introduce higher computational cost. Most of the existing approaches resolve the 3D pose estimation problem in two stages: first, a model is used to predict the 2D pose, then the 3D pose is obtained using different solutions that are based on the 2D information. For the single-view case, the 3D pose can be predicted through a depth regression model [33], or by memorization, matching the 3D with the 2D pose [9], or by using a probabilistic model [30]). The multi-view cases can project the 2D prediction to the 3D space with triangulation using the knowledge of geometry and camera positions [4]. Other methods directly predict the 3D pose without separately predicting the 2D pose: [23] simultaneously minimizes the 2D heatmaps and 3D pose, while [27] directly outputs a dense 3D volume with separate voxel likelihoods for each joint.

3. DHP19 Dataset

3.1. Data acquisition

Setup. Fig. 3 shows the dataset recording setup. The DHP19 dataset was recorded with four DAVIS cameras and simultaneous recording from the Vicon motion capture system, which provides the 3D position of the human joints. Recordings were made in a therapy environment in a recording volume of 2x2x2m^3. The Vicon setup was composed by ten Bonita Motion Capture (**BMC**) infrared (**IR**) cameras surrounding a motorized treadmill where the subjects performed the different movements. The high number of Vicon cameras is necessary in order to avoid marker occlusions. The BMC cameras emit 850 nm infrared light and sense the light reflected back from passive spherical markers located on the subject joints. The Vicon can attain a high sample rate (up to 200 Hz) and sub-millimeter precision. To collect the dataset, we choose a Vicon sampling rate of 100 Hz. The four DAVIS cameras used during the recording were suspended on the metallic frame, which also supported the BMC cameras (Fig. 3b)). The DAVIS cameras were arranged to provide almost 360-degrees coverage of the scene around the subject. The arrangement of all DAVIS and BMC cameras is shown in the design of Fig. 3c)-e). The DAVIS cameras were equipped with 4.5 mm focal-length lenses (Kowa C-Mount, f/1.4), and ultraviolet/infrared filters (Edmund Optics, 49809, cutoff 690 nm) to block most of the flashing Vicon illumination. We recorded only the DVS output since the host controller USB bandwidth was insufficient to capture all DVS and APS outputs simultaneously. However in a follow up study, APS and DVS outputs will be simultaneously collected to better compare CNN performance between event and frame based cameras. The motion capture system records the position of 13 labeled joints of the subject identified by the following markers: head, left/right shoulder, left/right elbow, left/right hand, left/right hip, left/right knees, and left/right foot. The output of Vicon cameras was recorded and processed using Vicon proprietary software (Nexus 2.6), that we used to visualize the markers, generate the skeleton structure and label the joints, as shown in Fig. 3d). We obtained the 3D pose ground-truth by approximating the marker positions as the true joint positions, without using a biomechanical model to calculate the joint centers.

Time synchronization. The DAVIS camera event timestamps are synchronized with the Vicon. The DAVIS cameras are daisy-chained using 3.5 mm audio cables that carry a 10 kHz clock, used by the camera logic to keep the internal timestamp counters synchronized. Camera1 (Fig. 3c)) is the master for the other cameras and receives a trigger input from the Vicon controller at the start and end of recording. These times are marked by two *special events* easily detectable in the DVS event stream. The Vicon *start* and *end* events allow aligning the camera recordings with the Vicon data.

Calibration. The motion capture system was calibrated using Vicon proprietary software and protocol for calibration.

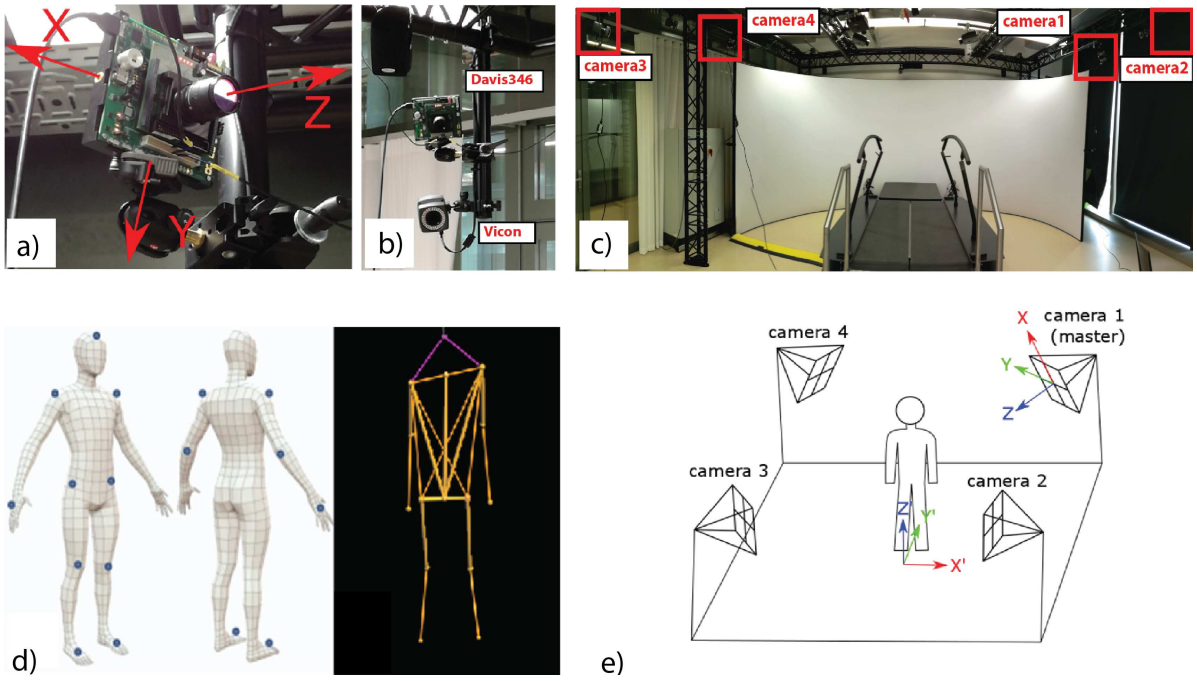

Figure 3. a-b) DAVIS and Vicon IR camera. c) Therapy environment setup at the Swiss Center for Clinical Movement Analysis. d) Vicon marker positions on the subject and skeleton representation. e) Schematic of the setup, with DAVIS master camera and Vicon origins.

Table 2. List of recorded movements

Session 1	Session 2	Session 3	Session 4	Session 5
1 - Left arm abduction	9 - Walking 3.5 km/h	15 - Punch straight forward left	21 - Slow jogging 7 km/h	27 - Wave hello left hand
2 - Right arm abduction	10 - Single jump up-down	16 - Punch straight forward right	22 - Star jumps	28 - Wave hello right hand
3 - Left leg abduction	11 - Single jump forwards	17 - Punch up forwards left	23 - Kick forwards left	29 - Circle left hand
4 - Right leg abduction	12 - Multiple jumps up-down	18 - Punch up forwards right	24 - Kick forwards right	30 - Circle right hand
5 - Left arm bicep curl	13 - Hop right foot	19 - Punch down forwards left	25 - Side kick forwards left	31 - Figure-8 left hand
6 - Right arm bicep curl	14 - Hop left foot	20 - Punch down forwards right	26 - Side kick forwards right	32 - Figure-8 right hand
7 - Left leg knee lift				33 - Clap
8 - Right leg knee lift				

To map the camera space to 3D space, each DAVIS camera was individually calibrated using images acquired from the APS output. The position of 38 Vicon markers was acquired in 8 different position and the 2D marker positions were manually labelled on the APS frames. The camera projection matrix P and the camera position C were calculated for each camera. P maps 3D world coordinates to image coordinates. It can be estimated using corresponding points in 3D and 2D space by solving the following system of equations:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (1)$$

where (u,v) defines the position of the 2D point on the camera plane, $p_{i,j}$ are the coefficients that need to be determined, with $p_{3,4}$ equal to 1, and (X, Y, Z) is the position of the 3D point in the world (Vicon) coordinate system. We marked the (u,v) positions in a set of images and solved the Eq. 1 system using least squares to obtain P for each camera. Once P is known, it is possible to calculate the camera position C. P can be defined as being made up of a 3x3 matrix (Q) and a 4th column (c_4). In this way, C is derived from Eq. 2:

$$P = (Q|c_4) \quad \Longrightarrow \quad C = Q^{-1}c_4 \quad (2)$$

3.2. Data description

Dataset contents. The DHP19 dataset contains a total of 33 movements recorded from 17 subjects (12 female and 5 male), between 20 and 29 years of age. The movements, listed in Table 2, are classified in: upper-limb movements (1, 2, 5, 6, 15-20, 27-33), lower-limb movements (3, 4, 7, 8, 23-26), and whole-body movements (9-14, 21, 22). The movements are divided into 5 sessions. Each movement is composed of 10 consecutive repetitions. We split the 17 subjects into 12 subjects for training and validation (9 female, 3 male), and 5 for testing (3 female, 2 male). The median duration of each 10-repetition file is 21 s. The median

DVS event rate per camera before noise filtering is 332 kHz.

DVS data format. The dataset contains only DVS data from the four DAVIS cameras. We adapted the standard DVS data format to the multi-camera setup case. We merged the streams of DVS events from each camera to ensure monotonic timestamp ordering, and included the identification (ID) number of each camera in the two least significant bits of the raw address. In this way, each event is represented by a tuple $e = (x, y, t, p, c)$. Where (x, y) is the address in the pixel array, t is the time information in microsecond resolution, p is the polarity of the brightness change, and c is the camera ID number. This arrangement makes it much easier to process all the DVS data together in single data files.

DVS events preprocessing. The raw event streams are preprocessed using a set of filters to clean them from the unwanted signal. In particular, we apply filters to remove the uncorrelated noise (background activity), to remove the hot pixels (pixels with abnormally low event thresholds), and to mask out spots where events are generated due to the infrared light emitted from the BMC cameras (not all the Near-IR signal from the BMC was removed by the IR filters). Fig. 1 shows representative samples from the application presented in Sec. 4. The left panels show DVS images from the four camera views and the right panels show the 3D Vicon ground truth skeleton synchronized with the DVS frame. The skeleton is generated using the mean value of the 3D joints in the time window of the accumulated frame.

3.3. Evaluation metric

For evaluation purposes we use the mean per joint position error (**MPJPE**), commonly used in HPE. MPJPE is equivalent to the average Euclidean distance between ground-truth and prediction, and can be calculated both in 2D and 3D space (respectively in pixel and mm) as:

$$\text{MPJPE} = \frac{1}{J}\sum_{i}^{J} \|x_i - \hat{x}_i\|, \tag{3}$$

where J is the number of the skeleton joints, and x_i and $\hat{x}_i$ are respectively the ground-truth and predicted position of the i-th joint in the world or image space.

4. DVS 3D human pose estimation

In this section we discuss our experiment with DHP19, demonstrating for the first time an application of HPE based on DVS data. In our experiment we use data from the two front views out of the four total DVS cameras (camera 2 and 3 in Fig. 1). Our choice is motivated by using the minimum number of cameras to make a 3D projection using triangulation. Future work will focus on using the two additional lateral cameras, more challenging due to a higher degree of self occlusion. We trained a single CNN on all the 33 movements for the 12 training subjects. Fig. 4 shows an overview of our approach to solve the problem of 3D HPE. In the proposed method we decompose the 3D pose estimation problem in 2D pose estimation based on CNN, and 2D-to-3D reconstruction using geometric information about the position of the cameras. First, a single CNN is trained on the two camera views. Then, we project each of the 2D predictions from the pixel space to the physical space through triangulation, knowing the projection matrices P and the camera positions C. The section is organized as follows: first, we discuss image and label preprocessing. Then, we introduce our method for 3D HPE, describing the CNN architecture, training setup, and prediction processing to obtain the final 3D human pose.

4.1. DVS frame generation

To leverage frame-based deep learning algorithms for event cameras, we need to turn the event stream representation into frames, referred to as DVS frames. Here we follow the strategy from [25] to generate DVS frames by accumulating a fixed number of events, which we call *constant count frames*. This allows us to have an adaptive frame rate that varies with the speed of the motion, and gives a constant amount of information in each frame. We fixed a number of 30 k events for the 4 DVS views (about 7.5 k events per single camera). Finally, the DVS frames are normalized in the range [0,255]. Following this procedure, about 87 k DVS frames were generated for each DVS camera.

4.2. Label preprocessing

Our CNN model predicts a set of 2D heatmaps, representing the probability of the joint presence at each pixel location, as proposed in [31]. To create the heatmaps from the 3D Vicon positions, we preprocess the Vicon labels as follows. Raw Vicon labels are collected at a sampling frequency of 100 Hz. In order to have input/output data pairs for training, the labels need to be temporally aligned to DVS frames. By knowing the DVS-frame initial and final event timestamps, we first take the Vicon positions at the closest sampling time, then we calculate the average position in that time window. We consider this average position as the 3D label of the corresponding DVS frame. Then, we use the projection matrices to project the 3D labels to 2D labels for each camera view, rounding to the nearest pixel position. The projected 2D labels represent the absolute position in pixel space. We create J heatmaps (one per joint, initialized to zero). For each 2D joint, the pixel corresponding to the (u,v) coordinate of the relative heatmap is set to 1. Finally, we smooth each heatmap using Gaussian blurring with a sigma of 2 pixels. This procedure is repeated for each joint and for each timestep.

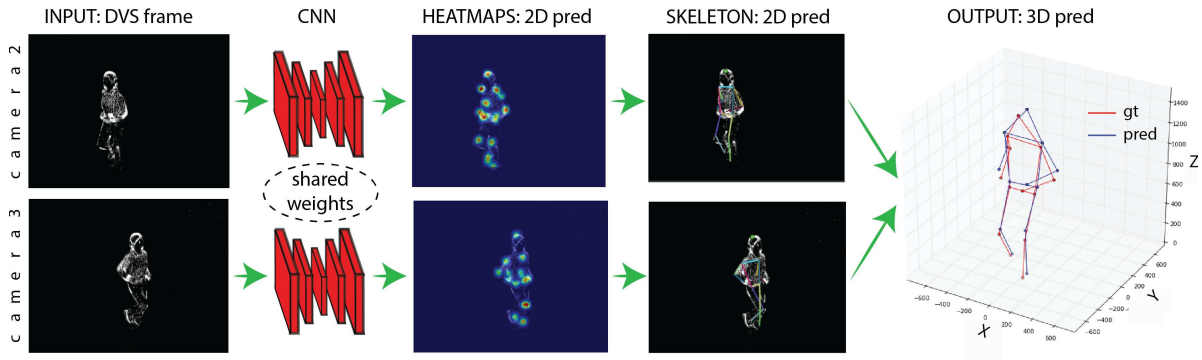

Figure 4. Overview of our proposed approach. Each camera view is processed by the CNN, joint positions are obtained by extracting the maximum over the 2D predicted heatmaps, and 3D position is reconstructed by triangulation.

4.3. Model

The proposed CNN has 17 convolutional layers (Table 3). Each layer has 3x3 filter size and is followed by Rectified Linear Unit (**ReLU**) activation. The DVS resolution is used as the CNN input resolution, and it is decreased with two max pooling layers in the first stages of the network. Then, it is recovered in later stages with two transposed convolution layers with stride 2. The convolutional layers of the network do not include biases. This architectural choice was motivated by an increase in the activation sparsity at a negligible decrease in performance. As discussed in Sec. 6, activation sparsity could be exploited for faster processing. The CNN has about 220 k parameters and requires 6.2 GOp/frame, where one Op is one multiplication or addition. In designing the CNN, we paid attention both to its prediction accuracy and computational complexity, to minimize model size for real-time applications.

4.4. Training

The CNN was trained for 20 epochs using RMSProp with Mean Square Error (MSE) loss and an initial learning rate of 1e-3. We applied the following learning rate schedule: 1e-4 for epochs 10 to 15, 1e-5 for epochs 15 to 20. The training took about 10 hours on an NVIDIA GTX 980 Ti GPU.

4.5. 2D prediction

The output of the CNN is a set of J feature maps, where J is the number of joints per subject. Each output pixel represents the confidence of the presence of the J-th joint, as done in [26]. For each output feature map, the position of the maximum activation is considered as the joint predicted position, while the value of the maximum activation is considered as the joint confidence. In this work we first evaluate the performance of the CNN instantaneous prediction. Then, we propose a simple method to keep into account past predictions, to account for immobile limbs. By looking at the DVS frames structure (Fig. 1), we observe that limbs that are static during the movement do not generate events: this can hence result in ambiguities in the pose estimation problem. The problem of static limbs could be mitigated by updating the CNN prediction of each joint at timestep T only when the confidence of that joint is above a certain threshold (confidence threshold), otherwise the CNN prediction from timestep (T-1) is retained. Despite its simplicity, this conditional update allows for an improvement in the 2D pose estimation performance, as discussed in Sec. 5.1.

4.6. 3D projection

For each camera of the two we used, we project the 2D position in 3D space using the inverse of the projection matrix P (Eq. 1). The 3D joint position is calculated as the point at minimum distance from the two rays passing through the back-projected point of each camera and the respective camera center C (Eq. 2).

5. Results

This section reports the results for HPE on the 5 test subjects. First, we present the CNN 2D pose prediction results, then those for the 3D pose estimation with geometrical projection of the CNN predictions. Finally, we present considerations in term of computational requirements and activation sparsity.

5.1. 2D pose estimation

Table 4 shows the 2D results on the test set, expressed as MPJPE (in pixel). We evaluate the prediction error both for instantaneous CNN prediction as well as for different values of confidence threshold, ranging from 0.1 to 0.5. We select a confidence threshold of 0.3, for which we observe a relative improvement of 7 % and 10 % in 2D MPJPE for camera 2 and 3 respectively. The CNN obtains a 2D MPJPE of about 7 pixels (camera 2: 7.18, camera 3: 6.87). Referring to Fig. 4, this average error in 2D joint position is about the size of the blobs in the HEATMAPS images.

Table 3. CNN architecture details. Changes in spatial resolution (Res.) are due to 2x2 max pooling (MP) or transposed convolution (TC) with stride 2. Dilation refers to the dilation rate in convolutional layers. The input is a constant-count DVS frame with shape 344x260x1.

Layer	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Stride	1	1	1	1	1	1	1	1	2	1	1	1	1	2	1	1	1
Dilation	1	1	1	1	2	2	2	2	1	2	2	2	2	1	1	1	1
Res.	MP			MP					TC					TC			
Output ch	16	32	32	32	64	64	64	64	32	32	32	32	32	16	16	16	13
Output H	130	130	130	65	65	65	65	65	130	130	130	130	130	260	260	260	260
Output W	172	172	172	86	86	86	86	86	172	172	172	172	172	344	344	344	344

Table 4. Test set 2D MPJPE (pixel) for the CNN, trained on the two frontal camera views (camera 2 and 3). In bold the selected confidence threshold (Conf. thr.) for 3D projection.

Conf. thr.	None	0.1	0.2	**0.3**	0.4	0.5
Camera 2	7.72	7.45	7.25	**7.18**	7.27	7.47
Camera 3	7.61	7.13	6.92	**6.87**	6.88	7.11

Table 5. 3D MPJPE (in mm) on the 5 test subjects on the 33 movements (M), over separate subjects (S) and whole sessions (Se). In bold the overall mean 3D MPJPE.

		Test Subject Number					Mean	Mean
Se	M	S1	S2	S3	S4	S5	M	Se
1	1	89.48	134.38	70.31	123.27	146.67	115.04	87.54
	2	60.57	77.95	78.14	128.37	147.50	99.65	
	3	68.71	105.79	92.83	71.32	84.37	84.65	
	4	68.20	89.58	70.70	78.96	80.59	78.35	
	5	78.20	104.76	119.21	119.79	94.02	103.29	
	6	128.18	125.78	114.38	127.40	105.62	121.06	
	7	62.25	78.36	70.10	85.49	76.24	74.97	
	8	58.77	77.34	67.20	77.17	79.17	71.95	
2	9	27.39	62.63	45.26	70.49	62.16	58.75	66.47
	10	41.24	48.16	49.79	75.34	129.85	82.23	
	11	68.34	55.46	54.86	79.53	113.04	80.53	
	12	29.34	51.05	44.13	76.88	55.78	53.57	
	13	45.97	50.94	50.79	70.10	56.42	55.56	
	14	31.25	52.38	46.27	69.88	61.70	54.21	
3	15	168.10	174.79	130.20	151.63	99.43	148.57	124.01
	16	127.25	139.56	121.22	147.53	116.49	135.92	
	17	72.54	157.16	90.42	115.44	99.98	111.35	
	18	109.84	120.48	117.14	114.36	209.16	131.46	
	19	67.88	124.43	91.76	107.28	111.82	106.92	
	20	70.26	100.34	73.32	112.47	90.92	98.28	
4	21	33.76	56.74	57.31	70.87	55.35	55.16	80.25
	22	56.67	73.21	70.30	100.49	73.29	76.23	
	23	99.74	150.30	96.40	97.72	102.59	111.66	
	24	106.96	130.38	118.13	104.67	85.26	112.49	
	25	91.33	105.81	162.77	81.98	135.30	118.00	
	26	76.87	88.54	140.02	83.47	139.78	104.67	
5	27	56.01	108.41	75.53	104.40	111.56	96.22	110.98
	28	73.75	108.68	65.46	126.57	95.58	101.32	
	29	68.66	*	91.40	150.94	120.34	110.59	
	30	78.14	103.33	99.08	157.88	102.11	112.44	
	31	64.82	98.76	118.08	168.75	104.66	110.69	
	32	94.29	132.31	95.17	146.34	130.89	123.59	
	33	93.27	90.47	169.01	161.56	108.25	122.93	
Mean		59.79	81.46	75.67	89.88	85.58	**79.63**	

*: video missing due to the absence of special event.

5.2. 3D pose estimation

Next we use the 2D pose estimates obtained from the CNN to calculate the 3D pose estimates. We calculate the 3D human pose by projecting the predicted 2D joint positions to 3D space with triangulation, as explained in Sec. 4.6. Table 5 reports the 3D MPJPE results for all subjects and movements, together with averages over single subject, movement, and session. Using a confidence threshold of 0.3, the average 3D MPJPE over all trained movements and test subjects is about 8 cm.In general, we notice that the best results are obtained for whole-body movements (movements 9-14, 21, 22, column Mean M in Table 5), for which all the human shape is visible in the DVS frames. Using a confidence threshold leads to improvements in the 3D prediction error (from 87.9 mm with no threshold, to 79.6 mm with a 0.3 threshold), but the absence of moving limbs in frames still represents a challenge for our model. This shortcoming becomes more evident when comparing the averages of whole-body movements against the other movements: 65.2 and 106.2 mm, respectively.

Table 6 compares our result on DHP19 with results from state-of-the-art models for multi-view settings. The significant differences across datasets, such as the type and range of movements, and subject orientation, do not allow for a direct comparison between the methods reported in Table 6. In particular, subjects in DHP19 keep the same orientation with respect to the cameras during all the movements. On the other hand, DHP19 provides a wider range of movements and subjects compared to the other datasets. As a general consideration, we observe that our prediction errors are within the range of current state-of-the-art methods. We believe this goes in favor of further exploring the DVS camera for HPE, and to develop new methods to take into account missing information due to non-moving parts.

6. Discussion

Presence of movement and its speed. The DVS microsecond time resolution provides a continuous temporal information not limited by a fixed frame rate, which can be advantageous for HPE by alleviating the motion blur present for fast movements (e.g. in MADS dataset [32]). In addition, static scenes generate only a few noise events and the CNN computation is not triggered, providing an adaptive frame rate that changes according to the speed of the movement being recorded. The frame-free, data-driven nature of the DVS event-stream means that the computational effort

Table 6. Qualitative comparison of 3D MPJPE (in mm) of our method on DHP19 and a variety of multi-view state-of-the-art models.

Dataset	Method	3D MPJPE	
		Walk	Box
HumanEva [29]	Amin et al. [4]	54.5	47.7
	Rhodin et al. [28]	74.9	59.7
	Elhayek et al. [11]	66.5	60.0
	Belagiannis et al. [6]	68.3	62.7
MADS [32]	Zhang et al. [32]	100-200	
DHP19	(Ours) All movements	79.6	
	(Ours) Whole-body	65.2	

is high only when needed, and at other times the hardware becomes idle and burns less power.

Immobile limbs. The problem of immobile limbs with DVS is partially mitigated by the introduction of the confidence threshold, but our results still show a significant gap in accuracy between partial-body and whole-body movements. Future work will focus on the pose estimate integration in time to better deal with the absence of limbs. Using model-based and learning approaches, such as constrained skeletons and Recurrent Neural Networks, on the instantaneous pose estimates provided by the CNN can constrain inference to possible pose dynamics.

Computational complexity. CNN power and latency also play a critical role for real-world applications. This section compares the requirements of our CNN with state-of-the-art CNNs that process RGB images, in terms of model parameters and operations. We compare our model to CNNs for 2D HPE because the 2D-to-3D component of our method is purely based on geometric properties, and does not include any learning. The DHP19 CNN requires 6.2 GOp/frame for an input resolution of 260x344 pixels, and has 220k parameters. For the same input resolution, a DeeperCut [15] part detector ResNet50 CNN [13] would requires about 20 GOp/frame and has 20M parameters. A Part Affinity Fields (**PAF**) 6-stage CNN [8] would require 179 GOp/frame and has 52M parameters. The DHP19 CNN has more than 100X fewer parameters and runs at least 3X faster than these other body part trackers. The discussed architectures are designed for different problems in the same context of HPE, hence a direct comparison is difficult. However, the reported numbers underline the importance of efficient CNN processing for real-time application. Additionally, by using constant-count DVS frames, the computation would be driven by movement, unlike conventional HPE systems that operate at constant frame rate.

Sparsity. Another way to reduce the latency of a CNN is to exploit the properties of the ReLU activation function, namely the clamping to zero of all the negative activations. The zero-valued activations of a layer do not contribute to the pre-activations of the next layer, and represent computation that can be avoided. Several hardware accelerators [3, 10] have been developed to take advantage of the activation sparsity by skipping over the zero activations. We calculate the activation sparsity, comparing our method with the PAF network. The DHP19 CNN has a sparsity of 89% (using a random sample of 100 DHP19 training images), while the PAF network sparsity is 72% (using images from the MS-COCO dataset [18]). The 2.5X sparser activation in the DHP19 CNN might result from the sparser DVS input. This result is encouraging in view of real-time HPE using custom hardware accelerators capable of exploiting sparsity.

7. Conclusion

The central contribution of this paper is the Dynamic Vision Sensor Human Pose dataset, which is the first dataset for 3D human pose estimation with DVS event cameras and labeled ground truth joint position data. We also provide the first deep network for human pose estimation based on the DVS input. Our proposed model is a proof of concept for demonstrating the usability of the dataset, but it also achieves joint accuracy within the range of multi-view state-of-the-art methods. Despite the limitations of the proposed approach due to static limbs, which will be addressed as future work, DVS cameras could enable more efficient human pose estimation towards real-time and power-constrained application. Furthermore, the high dynamic range of the DVS opens the possibility of HPE in embedded IoT systems that cannot use active illumination and must operate in all lighting conditions.

Acknowledgments

The authors thank all the people that volunteered for the collection of DHP19. Movement analysis was supported by the Swiss Center for Clinical Movement Analysis, SCMA, Balgrist Campus AG, Zurich. This work was funded by the EC SWITCHBOARD ETN (H2020 Marie Curie 674901), Samsung Advanced Inst. of Technology (SAIT), the University of Zurich and ETH Zurich.

References

[1] Event-based vision resources. https://github.com/uzh-rpg/event-based_vision_resources/. Last accessed: 2019-03-26. 2

[2] Vicon motion capture. https://www.vicon.com/. Last accessed: 2019-03-26. 2

[3] Alessandro Aimar et al. NullHop: A flexible convolutional neural network accelerator based on sparse representations of feature maps. *IEEE Transactions on Neural Networks and Learning Systems*, 30(3):644–656, 2019. 8

[4] Sikandar Amin, Mykhaylo Andriluka, Marcus Rohrbach, and Bernt Schiele. Multi-view pictorial structures for 3D human pose estimation. In *British Machine Vision Conference (BMVC)*, 2013. 3, 8

[5] Arnon Amir et al. A low power, fully event-based gesture recognition system. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7388–7397, 2017. 2

[6] Vasileios Belagiannis et al. 3D pictorial structures for multiple human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1669–1676, 2014. 8

[7] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240x180 130 dB 3 us Latency Global Shutter Spatiotemporal Vision Sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, Oct. 2014. 1, 2

[8] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7291–7299, 2017. 8

[9] Ching-Hang Chen and Deva Ramanan. 3D human pose estimation = 2D pose estimation + matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7035–7043, 2017. 3

[10] Yu-Hsin Chen, Tushar Krishna, Joel S Emer, and Vivienne Sze. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE Journal of Solid-State Circuits*, 52(1):127–138, 2017. 8

[11] Ahmed Elhayek et al. Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3810–3818, 2015. 3, 8

[12] Guillermo Gallego, Jon E. A. Lund, Elias Mueggler, Henri Rebecq, Tobi Delbruck, and Davide Scaramuzza. Event-based, 6-DOF Camera Tracking for High-Speed Applications. *arXiv:1607.03468 [cs]*, July 2016. arXiv: 1607.03468. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 8

[14] Yuhuang Hu, Hongjie Liu, Michael Pfeiffer, and Tobi Delbruck. DVS Benchmark Datasets for Object Tracking, Action Recognition and Object Recognition. *Neuromorphic Engineering*, 10:405, 2016. 2

[15] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision (ECCV)*, pages 34–50, 2016. 8

[16] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014. 2, 3

[17] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128 x 128 120 dB 15 us latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008. 1, 2, 3

[18] Tsung-Yi Lin et al. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755, 2014. 8

[19] Hongjie Liu et al. Combined frame- and event-based detection and tracking. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2511–2514, 2016. 2

[20] Iulia Lungu, Federico Corradi, and Tobi Delbruck. Live demonstration: Convolutional neural network driven by dynamic vision sensor playing RoShamBo. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017. 2

[21] Ana I. Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5419–5427, 2018. 2

[22] Dushyant Mehta et al. Monocular 3D human pose estimation in the wild using improved cnn supervision. In *International Conference on 3D Vision (3DV)*, 2017. 2, 3

[23] Dushyant Mehta et al. Vnect: Real-time 3D human pose estimation with a single RGB camera. *ACM Transactions on Graphics (TOG)*, 36(4):44, 2017. 3

[24] Dushyant Mehta et al. Single-shot multi-person 3D pose estimation from monocular RGB. In *International Conference on 3D Vision (3DV)*, 2018. 3

[25] Diederik P. Moeys et al. Steering a predator robot using a mixed frame/event-driven convolutional neural network. In *International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, pages 1–8, 2016. 5

[26] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision (ECCV)*, pages 483–499, 2016. 6

[27] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7025–7034, 2017. 3

[28] Helge Rhodin et al. General automatic human shape and motion capture using volumetric contour cues. In *European Conference on Computer Vision (ECCV)*, pages 509–526, 2016. 3, 8

[29] Leonid Sigal, Alexandru O. Balan, and Michael J. Black. HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1):4–27, 2010. 2, 3, 8

[30] Denis Tome, Christopher Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3D pose estimation from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2500–2509, 2017. 3

[31] Jonathan J. Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1799–1807, 2014. 5

[32] Weichen Zhang, Zhiguang Liu, Liuyang Zhou, Howard Leung, and Antoni B. Chan. Martial arts, dancing and sports dataset: A challenging stereo and multi-view dataset for 3D human pose estimation. *Image and Vision Computing*, 61:22–39, 2017. 2, 3, 7, 8

[33] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3D human pose estimation in the wild: a weakly-supervised approach. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 3