2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

Temporal Binary Representation for Event-Based Action Recognition

Simone Undri Innocenti
University of Florence
simone.undri@unifi.it

Federico Becattini
University of Florence
federico.becattini@unifi.it

Federico Pernici
University of Florence
federico.pernici@unifi.it

Alberto Del Bimbo
University of Florence
alberto.delbimbo@unifi.it

2020 25th International Conference on Pattern Recognition (ICPR) | 978-1-7281-8808-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPR48806.2021.9412991

Abstract—**In this paper we present an event aggregation strategy to convert the output of an event camera into frames processable by traditional Computer Vision algorithms. The proposed method first generates sequences of intermediate binary representations, which are then losslessly transformed into a compact format by simply applying a binary-to-decimal conversion. This strategy allows us to encode temporal information directly into pixel values, which are then interpreted by deep learning models. We apply our strategy, called Temporal Binary Representation, to the task of Gesture Recognition, obtaining state of the art results on the popular DVS128 Gesture Dataset. To underline the effectiveness of the proposed method compared to existing ones, we also collect an extension of the dataset under more challenging conditions on which to perform experiments.**

I. Introduction

Action Recognition has gained increasing importance in recent years, due to applications in several fields of research such as surveillance, human computer interaction, healthcare and automotive. Despite the significant steps forward made since the diffusion of deep learning, there are still challenges yet to be solved. Certain applications, for instance, have extremely high time constraints. This is the case when recognition must be performed from fast moving vehicles (e.g. drones or cars), or when the pattern to be recognized is extremely fast (e.g. eye glimpses). Indeed, standard RGB cameras might even fail to capture a rich enough signal to enable recognition due to low frame-rates and motion blur.

These limitations of RGB cameras have been addressed with event cameras. Event cameras, also known as neuromorphic cameras, are sensors that capture illumination changes, producing asynchronous events independently for each pixel. These sensors have several desirable properties such as high dynamic range, low latency, low power consumption, absence of motion blur and, last but not least, they operate at extremely high frequencies, generating events at a μs temporal scale. The output of an event camera therefore is highly different from the one of a regular RGB camera, making the applicability of computer vision algorithms not so straightforward. In particular, Deep Learning methods such as Convolutional Neural Networks (CNN), work with frames of synchronous data. Asynchronous events need to be aggregated into synchronous frames to be fed to a CNN.

Several event aggregation strategies have been proposed in literature, allowing the usage of frame-based algorithms [1], [2], [3], [4], [5]. These techniques however approximate the signal by quantizing time into aggregation intervals, yielding to a loss of information. The aggregation time can be lowered to limit this phenomena, but this will result in an extremely high number of frames to be processed, making real-time analysis prohibitive.

In this paper we present an event aggregation strategy named Temporal Binary Representation (TBR). Compared to existing strategies, TBR generates compact representations without losing information up to an arbitrarily small quantization time. In fact, we first aggregate events to generate intermediate binary representations with small quantization times and then losslessly combine sequences of intermediate representations into a single frame. This allows us to lower the amount of data to be processed while retaining information at fine temporal scales. TBR is specifically tailored for fast moving actions or gestures and can be directly used for training and evaluating standard CNNs. Indeed, we exploit two models based on Alexnet+LSTM and Inception 3D for action recognition, reporting state of the art results on the IBM DVS128 Gesture Dataset [6]. Furthermore, we highlight the benefits of the proposed strategy by collecting an extension of the dataset in more challenging scenarios, namely higher execution speed, multiple scales, camera pose and background clutter.

To summarize, the main contributions of this paper are the following:

- We propose a compact representation of event data dubbed Temporal Binary Representation, exploiting a conversion of binary event sequences into frames that encode both spatial and temporal information.
- Our formulation allows to tune information loss and memory footprint, making it suitable for real-time applications.
- We collected an extension of the popular DVS128 Gesture Dataset under challenging conditions, which we plan to release upon publication.

The paper is organized as follows: in Sec. II a literature review is reported to frame the work in the current state of the art; in Sec. III our Temporal Binary Representation is presented; in Sec. IV we provide an overview of the models used for classifying gestures; in Sec. V we present the dataset used for evaluating our approach and introduce the additional benchmark that we have collected; in Sec. VI we discuss the

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

training details; in Sec. VII and VIII we report the results of our approach; finally in Sec. IX we draw the conclusions.

II. Related Work

A. Action and Gesture Recognition

Several formulations have been adopted in literature for the task of action recognition. Early works [7], [8] have treated it as a classification task, while more recent works have provided a finer level of detail adding a temporal dimension (action detection) [9], [10], [11], [12], [13], [14], [15] or spatial information (action localization) [16], [17], [18], [19], [20].

Action detection aims at recognizing actions and determining their starting and ending points in untrimmed videos. These approaches are often based on temporal proposals [11], i.e. a set of frame intervals that are likely to contain a generic action, which are then classified or refined [12], [10]. This concept has been extended in the spatio-temporal action localization formulation, where the temporal boundaries of the action still need to be determined, but at the same time the actor needs to be precisely localized in each frame, as in an object detection task. The output of such systems is a spatio-temporal tube [16], [18], [21], i.e. a list of temporally adjacent bounding boxes enclosing the action.

Several works have been focusing on a specific subset of actions, referred to as gestures. Gestures can be divided into the three categories of body, hand and head gestures [22]. The interest in gestures often stems from the need to establish some form of interaction between humans and machines, which indeed can happen interpreting human behaviors [23]. To reduce the reaction time to observed gestures, sensors with high frame-rate have been exploited [24]. Of particular interest is the usage of event cameras, which have been largely used for gesture recognition in the recent years [25], [26], [27], [6], [28], [29], [3], [30]. Some approaches rely on architectures specifically tailored to handle event data, such as spiking neural networks, which however require specialized hardware to be implemented [31], [29], [27]. Most approaches, however, in order to exploit traditional computer vision algorithms, adopt an event aggregation strategy that allows the conversion of streams of asynchronous events into a set of synchronous frames. Most of these approaches, though, perform a temporal quantization in the form of histograms [3] or event subsampling [26]. To avoid information loss, the bins into which events are quantized can be shrinked, with the side effect of generating a large amount of data that has to be processed. Differently from these works, we propose an aggregation strategy that is lossless up to an arbitrarily small time interval. Our proposed approach in fact compacts several representations in a single frame, allowing to generate less data without discarding information.

III. Event Representation

Events generated by an event camera are temporally and spatially localized respectively by a timestamp t and pixel coordinates x and y. Each event is also associated to a polarity $p \in \{-1, +1\}$, indicating the sign of the pixel illumination change. The output of an event camera is therefore a stream of tuples $E = (t, x, y, p)$. To make events interpretable by standard Computer Vision algorithms, they must be aggregated into frames. In general, an aggregation algorithm is a function that maps asynchronous events into a a stream of synchronous frames. Each generated frame f^i aggregates all the events in the interval $[t^i; t^i + \Delta t]$ spanning from an initial timestamp t^i and covering a temporal extent Δt, known as accumulation time.

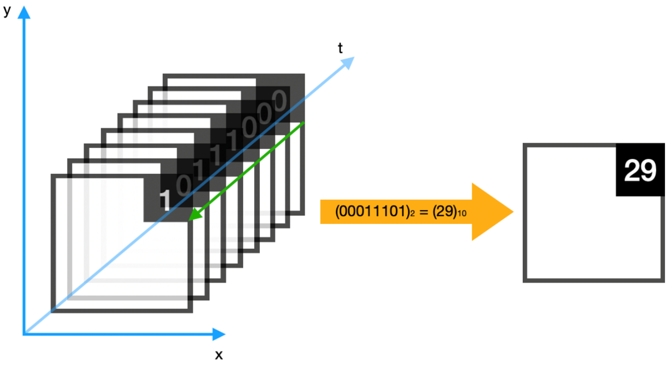

Fig. 1. Temporal Binary Representation. Events are first stacked together into intermediate binary representations which are then grouped into a single frame thanks to a binary to decimal conversion.

A. Temporal Binary Representation

Given a fixed Δt, we build an intermediate binary representation b^i by simply checking the presence or absence of an event for each pixel during the accumulation time. The value in position (x, y) is obtained as $b^i_{x,y} = \mathbb{1}(x, y)$, where $\mathbb{1}(x, y)$ is an indicator function returning 1 if an event is present in position (x, y) and 0 otherwise.

We then consider N temporally consecutive binary representations by stacking them together into a tensor $B \in \mathbb{R}^{H \times W \times N}$. Each pixel can be considered as a binary string of N digits $[b^0_{x,y} \; b^1_{x,y} \; ... \; b^{N-1}_{x,y}]$ with the most significant digit corresponding to the most recent event. We then convert into a decimal number each binary string, as shown in Fig. 1. This procedure allows us to compact the representation of N consecutive accumulation times into a single frame without any loss of information. The frame is then normalized in $[0, 1]$, dividing its values by N. We refer to this event representation as Temporal Binary Representation (TBR).

Compared to standard event aggregation strategies that generate a single frame for each Δt, TBR reduces the memory footprint by a factor of N. This also leads to less data to be processed by Computer Vision algorithms, enabling time-constrained applications. At the same time, the accumulation time can be significantly reduced to capture events at finer temporal scales, without increasing the total number of frames.

IV. Model

We adopt our Temporal Binary Representation for event camera data to the task of Action Recognition. To process frames, we use two different architectures.

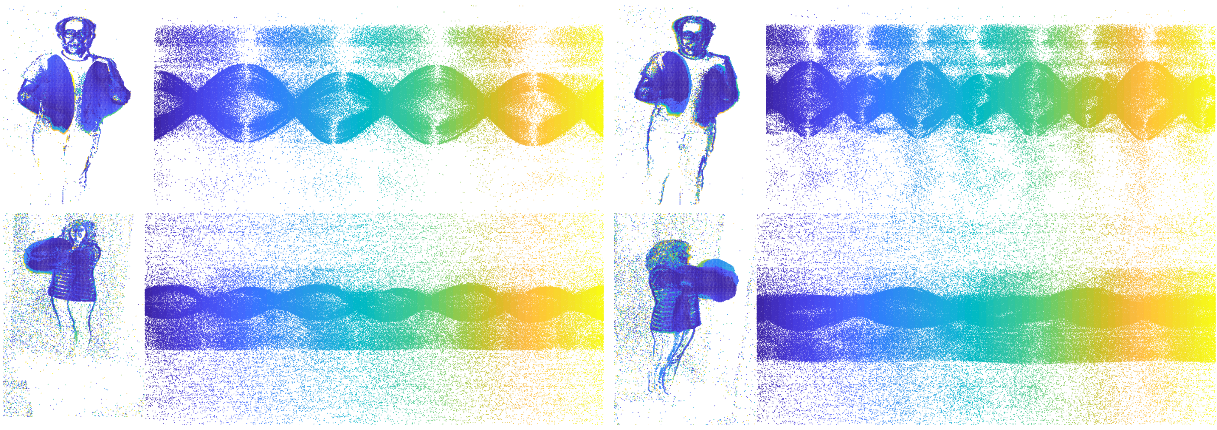

Fig. 2. Samples from the MICC-Event Gesture Dataset. Slow and fast execution of the action *air drum* (first row) and different scale and orientation of the action *arm roll* (second row). A 1 second snippet is shown for each sample, where events are color-coded according to the timestamps from blue (start - 0s) to yellow (end - 1s). The actors are shown both frontal (left) and sideways (right).

First, we combine a Convolutional Neural Network to extract spatial features with a Recurrent Neural Network to process sequences of frames. As a feature extractor, we train an AlexNet [32], replacing the final fully connected layers using a single layer with 512 neurons. The output for each frame in the sequence is directly fed to a Long Short Term Memory (LSTM) with 2 layers with hidden dimension 256 each. Finally, a fully connected layer with softmax activation performs the classification.

The second model that we adopt is the Inception 3D model [33], a state of the art architecture widely adopted with RGB data for action recognition. Based on Inception-V1 [34], the model relies on inflated 3D convolutions by adding a third dimension to filters and pooling kernels to learn spatio-temporal feature extractors. The model originally has two separate streams for RGB and Optical Flow data. Here we simply remove one branch and retrain the model with event camera data aggregated with TBR.

To process videos, we follow two different approaches, depending on the network. For the AlexNet+LSTM model we simply feed the whole sequence of frames to the model and collect the final output. With Inception 3D instead, we use as input non-overlapping blocks of F frames stacked together, which are independently evaluated. To provide a final classification for the whole video, we adopt a majority-voting strategy among predictions for each block.

V. Datasets

We train our model on the the IBM DVS128 Gesture Dataset [6]. The dataset contains a total of 1342 hand gestures with a variable duration spanning from approximately 2 to 18 seconds (6 seconds on average). Gestures are divided in 10 classes plus an additional random class for unknown gestures. Each of these actions are performed by 29 subjects under different illumination conditions (natural, fluorescent and led lights). The data is acquired using a DVS128 camera, i.e. an event camera with a sensor size of 128 × 128 pixels [35]. We follow the split proposed by the authors, comprising 23 subjects for training and 6 for validation.

To increase the variability of the DVS128 Gesture Dataset we recorded an additional test benchmark using a Prophesee GEN 3S VGA-CD event camera[1]. The camera has a sensor with a higher resolution of 640 × 480 pixels (VGA). The recorded actions still belong to the 11 classes of the DVS128 dataset but are performed under more challenging conditions. In particular, the actors were asked to perform the actions at different speeds, in order to demonstrate the capacity of event cameras to capture high speed movements. In addition the actions have been recorded at different scales and camera orientations and also under uneven illumination which is likely to cast shadows on the body and the surroundings, generating spurious events. The dataset was recorded by 7 different actors of different age, height and gender for a total of 231 videos. All the videos are used for testing, still using the DVS128 Gesture Dataset as training set. We refer to the newly collected data as the MICC-Event Gesture Dataset, which will be released upon publication. In Fig. 2 a few samples from the dataset are shown, highlighting the different execution speeds, scales and orientations at which actions are recorded.

VI. Training

We train the models using the SGD optimizer with momentum. We use a learning rate equal to 0.01, which is then decreased to 0.001 after 25 epochs. As loss we adopt the Binary Cross-Entropy Loss, regularized with weight decay. Overall, the training of Inception 3D took 13 hours on an NVIDIA Titan Xp, while AlexNet+LSTM required approximately 30 hours.

For the DVS128 Gesture Dataset, to make the frames compatible with the input layer of the models, we apply a zero-padding up to 227×227 for AlexNet+LSTM and 224×224 for Inception 3D. For the MICC-Event Gesture Dataset instead, which is recorded with the higher resolution of 640×480, we

[1] https://www.prophesee.ai/event-based-evk/

TABLE I

RESULTS ON THE DVS128 GESTURE DATASET.

	10 classes	11 classes
Time-surfaces [25]	96.59	90.62
SNN eRBP[26]	-	92.70
Slayer [27]	-	93.64
CNN [6]	96.49	94.59
Space-time clouds [28]	97.08	95.32
DECOLLE [29]	-	95.54
Spatiotemporal filt. [3]	-	97.75
RG-CNN [30]	-	97.20
Ours - AlexNet+LSTM	97.50	97.73
Ours - Inception3D	**99.58**	**99.62**

perform a central crop of 350 × 350 pixels and then reshape it to 128 × 128 to match the size of DVS128. Reshape is done with Nearest Neighbor interpolation to a avoid unwanted artifacts that may introduce noise in the event representation. Frame values are normalized in $[-1;1]$ before being fed to the models. During training we also perform data augmentation applying random scaling, translation and rotation.

VII. EXPERIMENTS

In Tab. I we report the results on the DVS128 Gesture Dataset for the two models AlexNet+LSTM and Inception 3D, trained with frames generated by our Temporal Binary Representation. The results are compared with state of the art approaches. Following prior work, we report the classification accuracy both including and excluding the *Other Gesture* class, respectively referred to as ”10 classes” and ”11 classes”.

In our models, events are aggregated with the proposed Temporal Binary Encoding, stacking $N = 8$ binary representations with an accumulation time $\Delta t = 2.5ms$. Therefore, we use an 8 bit representation for each pixel, covering 20 ms with each frame. It is important to notice that this allows the model to observe events without any loss of information up to a precision of 2.5 ms, even if a single frame stores data covering an 8 times bigger time interval. Since the Inception 3D model takes as input chunks of videos as a tensor of temporally stacked frames, we feed to the model chunks of 500 ms, i.e. chunks of 25 frames encoded with TBR. With classic event aggregation strategies that use the same Δt of 2.5 ms, this would lead to 200 frames per chunk, increasing considerably the computational burden.

Overall, the Inception 3D model achieves the best results, reporting approximately a 2% improvement respect to AlexNet+LSTM. Interestingly, both our architectures are capable to obtain a perfect classification of the *Other Gesture* class, making the accuracy in the 11 classes settings higher than the 11 classes one. This behavior is the opposite compared to the baselines that adopt the 10 classes setting, which consistently lowers the accuracy.

To better assess the benefits of adopting our Temporal Binary Representation, we report results on the MICC-Event Gesture Dataset. We use the whole dataset for testing the Inception 3D model, which is trained on DVS128. To provide a comparison with other approaches, we have trained 2 baseline variants using event aggregation strategies from the literature: *Polarity* [1] and *Surface of Active Events* [36].

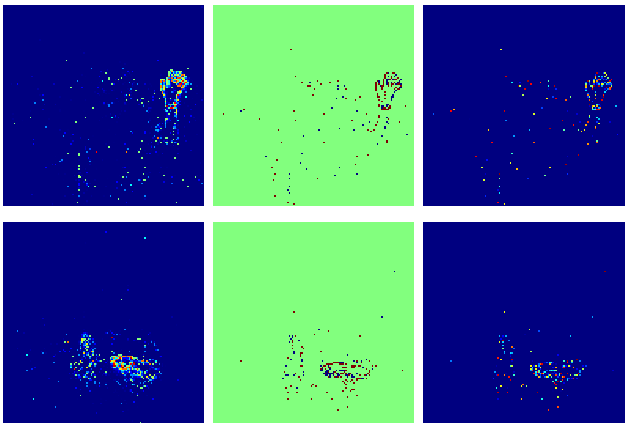

Fig. 3. Events aggregated with our Temporal Binary Representation (left), Polarity [1] (middle) and Surface of Active Events [36] (right). All three representations are made using an accumulation time $\Delta t = 2.5ms$.

TABLE II

RESULTS ON THE DVS128 GESTURE DATASET AND THE MICC-EVENT GESTURE DATASET FOR INCEPTION 3D TRAINED WITH THREE DIFFERENT AGGREGATION STRATEGIES: TBR (OURS), POLARITY [1] AND SAE [36].

	TBR (ours)	Polarity	SAE
DVS128 Gesture Dataset	**99.62**	98.86	98.11
MICC-Event Gesture Dataset	**73.16**	68.40	70.13

The *Polarity* [1] approach simply assigns a different value to events with different polarities. Therefore, the final representation is an image I_p, where each pixel (x, y) is given by:

$$I_p(x,y) = \begin{cases} 0, & \text{if event polarity is negative} \\ 0.5, & \text{if no events happen in } \Delta\text{t} \\ 1, & \text{if event polarity is positive} \end{cases} \quad (1)$$

If multiple events are detected in the accumulation time, the most recent one is considered.

The *Surface of Active Events* (SAE) [36] instead, for each pixel measures the time between the last observed event and the beginning of the accumulation time t_0. The values are then normalized between 0 and 255, similarly to TBR with 8 bits. Polarity is discarded. The representation I_{SAE} is obtained as:

$$I_{SAE}(x,y) = 255 \times \left(\frac{\text{t}_\text{p} - \text{t}_0}{\Delta\text{t}}\right) \quad (2)$$

Samples using TBR, Polarity and SAE are shown in Fig. 3.

In Tab. II we show the results obtained by Inception 3D trained with the three different aggregation strategies. All three strategies are used with an accumulation time $\Delta t = 2.5ms$. We also report the results on the original DVS128 Gesture Dataset test set obtained by our model with the baseline aggregation strategies. Interestingly, on DVS218 the three variants still obtain higher performances than the existing methods from the literature reported in Tab. I. This confirms the choice of Inception 3D, which proves to be suitable for the task of action/gesture recognition using event data.

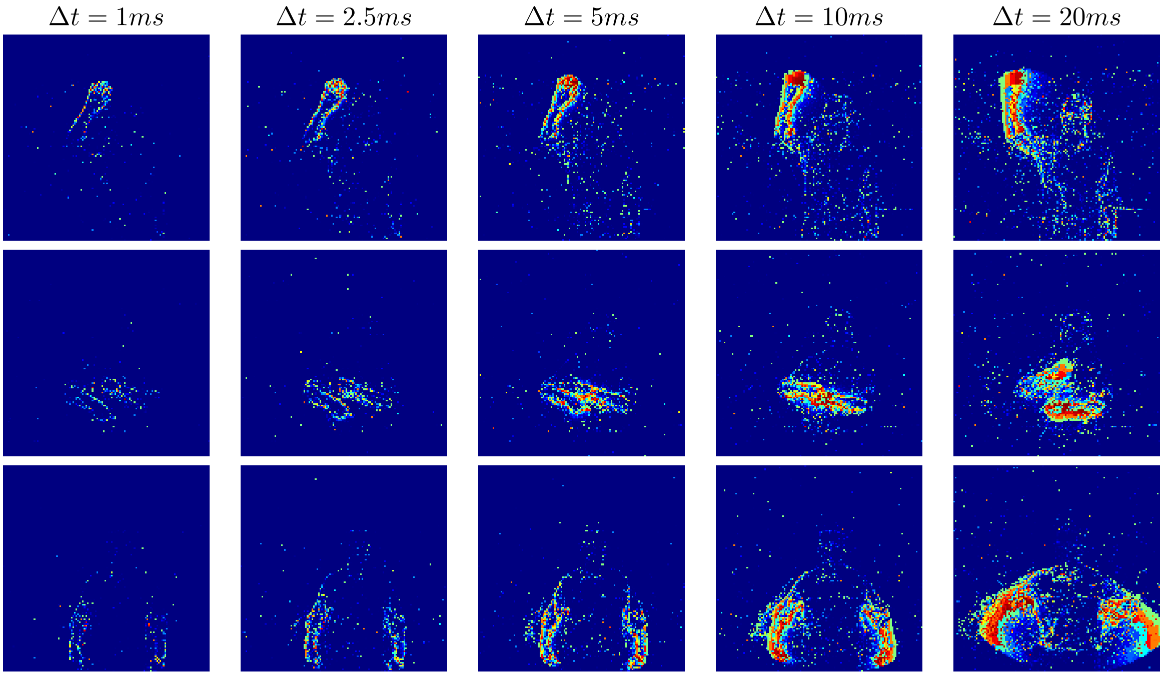

Fig. 4. Temporal Binary Representations with different accumulation times Δt with a number of bits $n = 8$. Each frame represents all the events in the interval $[0; n \times \Delta t]$. Three different gestures are shown: *Right Hand Clockwise* (top); *Arm Roll* (middle); *Other Gesture* (bottom). Pixels are color-coded according to intensity, from 0 (blue - no events) to 255 (red - an event registered for each bit of the representation).

The results on the MICC-Event Gesture Dataset overall are much lower due to the challenging scenarios that we have collected. However, the gap between the proposed aggregation strategy and the baselines increases considerably, suggesting that the Temporal Binary Representation is capable of representing event data more effectively. At the same time, since we are using $N = 8$ bits, TBR generates 8 times less data to be processed since N frames are losslessly condensed into a single representation.

VIII. Ablation Studies

We perform a series of ablation studies, showing the performance of the proposed method varying the parameters of the Temporal Binary Representation strategy. In particular, we observe how the accuracy of the system is affected when varying the accumulation time Δt, the number of bits used for the binary representation and the length of the video chunk fed to the Inception 3D model.

A. Accumulation time

Varying the accumulation time Δt, we can adjust the temporal quantization made by TBR. Higher accumulation times lead to more compact representations, which however carry less information. It can be seen from Fig. 5 that this information loss comes with a drop in accuracy for accumulation times bigger than 2.5 ms. Interestingly, lowering Δt beneath this threshold does not bring any improvement for the task at hand. In the plot, the best result from the state of the art [3], is shown as reference.

Fig. 4 shows samples of Temporal Binary Representations for different accumulation times. Especially for sufficiently high Δt, both the spatial and temporal nature of the encoding appears clearly visible.

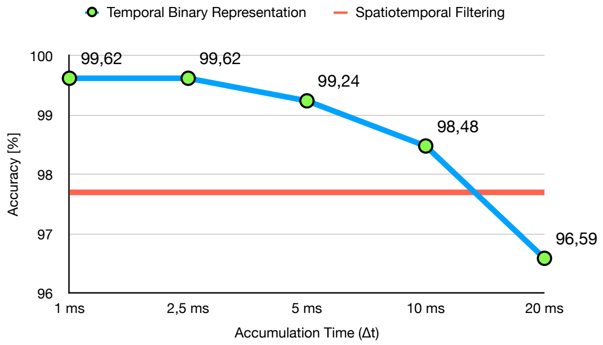

Fig. 5. Accuracy of Inception 3D on the DVS 128 Gesture Dataset, varying the accumulation time Δt. The best results from the state of the art [3] is also shown as reference.

B. Number of bits

Along with Δt, the number of bits N used for the proposed Temporal Binary Representation, defines how much information gets condensed into a single frame. Fig. 6 shows the accuracy of Inception 3D on the DVS128 Gesture Dataset using $N = 4, 8, 16$. Similarly to Δt, when N becomes too small, the accuracy of the model saturates. Throughout the paper we have taken $N = 8$ bits as reference for building our representations since it offers a trade-off between accuracy and data compactness. Furthermore, the choice of $N = 8$ simplifies data storage since events can be saved as unsigned integers grayscale images with lossless compression.

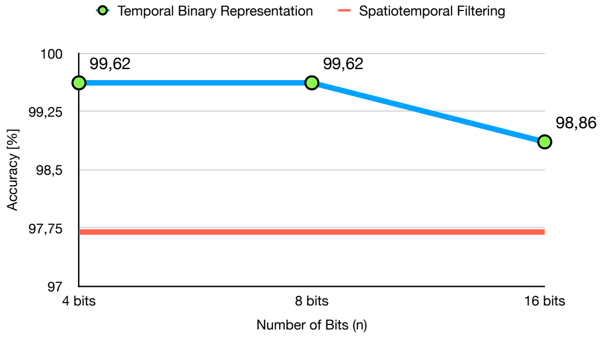

Fig. 6. Accuracy of Inception 3D on the DVS 128 Gesture Dataset, varying the number of bits for the Temporal Binary Representation. The best results from the state of the art [3] is also shown as reference.

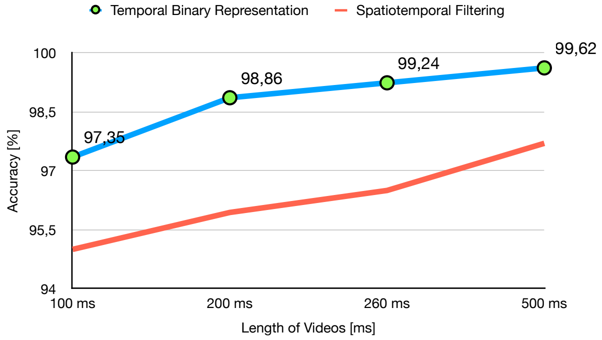

Fig. 7. Accuracy of Inception 3D on the DVS 128 Gesture Dataset, varying the chunk size fed to Inception 3D. The best results from the state of the art [3] is also shown as reference.

C. Chunk length

Here we vary the length of the chunks fed to Inception 3D. Since the model exploits 3D inflated convolutions, it can process multiple frames concatenated together, therefore taking into account the temporal dimension. In the case of TBR, the temporal dimension is already encoded covering a timespan of $N \times \Delta t$. By staking frames together we are extending the observation timespan by a factor equal to the number of frames. This setting is equivalent to the one adopted in [3], where the classifier performs a majority voting after having observed several chunks of various dimensions. In Fig. 7, we report the results for both methods, varying the chunk length from 100 ms to 500 ms. For our Temporal Binary Encoding we use $\Delta t = 2.5ms$ and $N = 8$, hence covering with each frame a temporal interval of 20 ms. The accuracy of the system improves when the chunk length increases, up to 500 ms. We did not observe significant improvements when increasing it further by adding more frames. It has to be noted however that increasing the chunk length will also increase the latency of the model, since a longer part of the gesture needs to be observed before emitting the first classification.

IX. Conclusions

In this paper we have presented an accumulation strategy called Temporal Binary Representation for converting the output of event cameras from raw events to frames, making them processable by Computer Vision algorithms. The proposed approach generates highly compact data, thanks to a lossless conversion of intermediate binary representations into a single decimal one. The effectiveness of the proposed approach has been validated on the commonly used DVS128 Gesture Dataset, reporting state of the art results. In addition a new test benchmark for event-based gesture recognition has been collected and will be publicly released.

Acknowledgments

This work was partially supported by the Italian MIUR within PRIN 2017, Project Grant 20172BH297: I-MALL - improving the customer experience in stores by intelligent computer vision.

References

[1] A. Nguyen, T.-T. Do, D. G. Caldwell, and N. G. Tsagarakis, “Real-time 6dof pose relocalization for event cameras with stacked spatial lstm networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[2] S. Miao, G. Chen, X. Ning, Y. Zi, K. Ren, Z. Bing, and A. Knoll, “Neuromorphic vision datasets for pedestrian detection, action recognition, and fall detection,” *Frontiers in neurorobotics*, vol. 13, p. 38, 2019.

[3] R. Ghosh, A. Gupta, A. Nakagawa, A. Soares, and N. Thakor, “Spatiotemporal filtering for event-based action recognition,” *arXiv preprint arXiv:1903.07067*, 2019.

[4] M. Cannici, M. Ciccone, A. Romanoni, and M. Matteucci, “A differentiable recurrent surface for asynchronous event-based data,” in *European Conference on Computer Vision*. Springer, 2020, pp. 1–17.

[5] ——, “Asynchronous convolutional networks for object detection in neuromorphic cameras,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[6] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza *et al.*, “A low power, fully event-based gesture recognition system,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7243–7252.

[7] J. K. Aggarwal and M. S. Ryoo, “Human activity analysis: a review,” *ACM Computing Surveys*, vol. 43, no. 3, pp. 16:1–16:43, 2011.

[8] R. Poppe, “A survey on vision-based human action recognition,” *Image and Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.

[9] A. Gaidon, Z. Harchaoui, and C. Schmid, “Temporal localization of actions with actoms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2782–2795, 2013.

[10] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, “End-to-end learning of action detection from frame glimpses in videos,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2678–2687.

[11] F. C. Heilbron, J. C. Niebles, and B. Ghanem, “Fast temporal activity proposals for efficient detection of human actions in untrimmed videos,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1914–1923.

[12] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, “Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[13] V. Escorcia, C. D. Dao, M. Jain, B. Ghanem, and C. Snoek, “Guess where? actor-supervision for spatiotemporal action localization,” *Computer Vision and Image Understanding*, vol. 192, p. 102886, 2020.

[14] D. Liu, T. Jiang, and Y. Wang, “Completeness modeling and context separation for weakly supervised temporal action localization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1298–1307.

[15] P. X. Nguyen, D. Ramanan, and C. C. Fowlkes, “Weakly-supervised action localization with background modeling,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5502–5511.

[16] G. Gkioxari and J. Malik, "Finding action tubes," in *IEEE International Conference on Computer Vision*, 2015, pp. 759–768.

[17] X. Peng and C. Schmid, "Multi-region two-stream r-cnn for action detection," in *European Conference on Computer Vision*, 2016, pp. 744–759.

[18] S. Saha, G. Singh, M. Sapienza, P. H. Torr, and F. Cuzzolin, "Deep learning for detecting multiple space-time action tubes in videos," in *British Machine Vision Conference*, 2016.

[19] G. Singh, S. Saha, M. Sapienza, P. Torr, and F. Cuzzolin, "Online real-time multiple spatiotemporal action localisation and prediction," in *IEEE International Conference on Computer Vision*, 2017.

[20] F. Becattini, T. Uricchio, L. Seidenari, L. Ballan, and A. Del Bimbo, "Am i done? predicting action progress in videos," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2020.

[21] G. Cuffaro, F. Becattini, C. Baecchi, L. Seidenari, and A. Del Bimbo, "Segmentation free object discovery in video," in *European Conference on Computer Vision*. Springer, 2016, pp. 25–31.

[22] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, 2007.

[23] H. Liu and L. Wang, "Gesture recognition for human-robot collaboration: A review," *International Journal of Industrial Ergonomics*, vol. 68, pp. 355–367, 2018.

[24] T. Sato, K. Fukuchi, and H. Koike, "Ohajiki interface: flicking gesture recognition with a high-speed camera," in *International Conference on Entertainment Computing*. Springer, 2006, pp. 205–210.

[25] J.-M. Maro, S.-H. Ieng, and R. Benosman, "Event-based gesture recognition with dynamic background suppression using smartphone computational capabilities," *Frontiers in Neuroscience*, vol. 14, p. 275, 2020.

[26] J. Kaiser, A. Friedrich, J. Tieck, D. Reichard, A. Roennau, E. Neftci, and R. Dillmann, "Embodied neuromorphic vision with event-driven random backpropagation," *arXiv preprint arXiv:1904.04805*, 2019.

[27] S. B. Shrestha and G. Orchard, "Slayer: Spike layer error reassignment in time," in *Advances in Neural Information Processing Systems*, 2018, pp. 1412–1421.

[28] Q. Wang, Y. Zhang, J. Yuan, and Y. Lu, "Space-time event clouds for gesture recognition: from rgb cameras to event cameras," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 1826–1835.

[29] J. Kaiser, H. Mostafa, and E. Neftci, "Synaptic plasticity dynamics for deep continuous local learning (decolle)," *Frontiers in Neuroscience*, vol. 14, p. 424, 2020.

[30] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, and Y. Andreopoulos, "Graph-based spatial-temporal feature learning for neuromorphic vision sensing," *arXiv preprint arXiv:1910.03579*, 2019.

[31] P. O'Connor, D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer, "Real-time classification and sensor fusion with a spiking deep belief network," *Frontiers in neuroscience*, vol. 7, p. 178, 2013.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[33] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[35] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change," in *2006 IEEE International Solid State Circuits Conference-Digest of Technical Papers*. IEEE, 2006, pp. 2060–2069.

[36] E. Mueggler, C. Bartolozzi, and D. Scaramuzza, "Fast event-based corner detection," 2017.