A Novel Approach for Neuromorphic Vision Data Compression based on Deep Belief Network

Sally Khaidem  
Indian Institute of Technology Madras  
Chennai, TN, India  
ee20d041@smail.iitm.ac.in

Abhipraay Nevatia  
Indian Institute of Technology Madras  
Chennai, TN, India  
ee20b007@smail.iitm.ac.in

Mansi Sharma  
Indian Institute of Technology Madras  
Chennai, TN, India  
mansisharma@ee.iitm.ac.in

Figure 1: DAVIS [23] dataset with different types motion (translation, rotation and 6-dof) (a) Shapes (b) Boxes (c) Poster.

ABSTRACT
A neuromorphic camera is an image sensor that emulates the human eyes capturing only changes in local brightness levels. They are widely known as event cameras, silicon retinas or dynamic vision sensors (DVS). DVS records asynchronous per-pixel brightness changes, resulting in a stream of events that encode the time, location, and polarity of brightness change. DVS consumes little power and can capture a wider dynamic range with no motion blur and higher temporal resolution than conventional frame-based cameras. Despite yielding a lower bit rate compared to conventional video capture, the present approach of event capture demonstrates enhanced compressibility. Hence, we introduce a novel deep learning-based compression methodology tailored for event data. The proposed technique employs a deep belief network (DBN) to condense the high-dimensional event data into a latent representation, which is subsequently encoded utilising an entropy-based coding method. Notably, our proposed scheme represents one of the initial endeavours to integrate deep learning methodologies for event compression. It achieves a high compression ratio while maintaining good reconstruction quality outperforming state-of-the-art event data coders and other lossless benchmark techniques.

1 INTRODUCTION
In conjunction with the brain’s cognitive processing, the sense of vision takes a primary role among human sensory modalities in perceiving the surrounding environment and acquiring new knowledge. “Silicon Retina” [19] mimics the neural architecture of human eyes and reveals a new, powerful way of computations, sparking the emerging field of neuromorphic engineering. Bio-inspired novel sensors such as Dynamic Vision Sensors (DVS) [17] measure intensity changes asynchronously rather than capturing intensity images at a fixed rate. As a result, it generates a stream of events that encodes the time, location, and polarity of brightness changes, where the data rate depends on scene complexity and camera speed. When compared to traditional cameras, DVS have superior properties. They have a very high dynamic range (140 dB versus 60 dB), no motion blur, and measurements with latency on the order of microseconds. DVS devices, such as Dynamic and Active-pixel Vision Sensor (DAVIS) [5] and Asynchronous Time-based Image Processing
Sensor (ATIS) [24] are a viable alternative in challenging conditions for standard cameras, such as high-speed high-dynamic-range motion photography, robotic automation, and intelligent surveillance [13, 25, 28, 31].

The neuromorphic silicon technology uses Address Event Representation (AER) [6], a communication protocol for transferring spikes events between bio-inspired chips. A tuple \((X, Y, p, t)\) represents each event, where \(X\) and \(Y\) denote the location of the event at a particular timestamp \(t\) with polarity \(p\) indicating an increase or decrease in event brightness. Each tuple is represented by 64 bits, with the timestamp being 32 bits and the remaining three fields being 32 bits. The goal is to gather helpful information from event data and utilize it for processing.

DVS acquire information asynchronously and sparsely, with high temporal resolution and low latency. Hence, the temporal aspect, particularly latency, is critical in the event data processing. The output stream cannot use traditional vision algorithms since it is a series of asynchronous events rather than actual intensity images. Therefore, development of new algorithms that take advantage of the sensor’s high temporal resolution and asynchronous nature is necessary. There are two types of algorithms based on the processing number of events at the same time. The first approach operates on an event-by-event basis, in which the system’s state changes upon the occurrence of a single event, resulting in minimal latency. The second approach involves latency because it operates on groups or packets of events. It can still provide a system state update upon the occurrence of each event if the window moves by one event at a time. The data storage and transmission bandwidth limitation for onboard DVS processing is an open challenge and requires immediate solutions. Spike coding [3] is a dedicated lossless compression strategy that exploits event data’s time-series and asynchronous nature. It follows a cube-based coding framework where the spike sequence is divided into multiple macro-cubes and encoded accordingly. Entropy-based coding strategies like Huffman [30] and Arithmetic [2] can effectively encode DVS data by treating each spike event field as an input symbol. Existing lossless coding schemes such as dictionary-based [1, 7, 8] and fast-integer [4, 15] encoders can also compress the DVS data after converting the spike events into a multivariate stream of integers.

The applications of DVS range from self-driving cars [20] to robotics [26] and drones [22]. Applications such as coordinating multiple intelligent vehicles (IoV) (cars, drones, etc.) having onboard processing constraints require real-time data sharing and feedback. In contrast to conventional sensing techniques, neuromorphic sensing exhibits an inherent ability to achieve compression. Such supplementary compression of event data holds distinct advantages for efficient transmission within the Internet of Things (IoT) and the Internet of Vehicles (IoV) domains. As a contribution to the field, this paper introduces a novel approach designed explicitly for compressing Dynamic Vision Sensor (DVS) data, employing the deep learning algorithm known as Deep Belief Network (DBN) [21]. Figure 3 depicts the complete workflow of event compression. The comprehensive event stream transforms into a dimensionally reduced latent representation facilitated by multiple code layer blocks, employing the Deep Belief Network (DBN). Recurring information is embedded within the resulting compact latent code blocks, rendering them well-matched for lossless symbol-based encoders. Consequently, the latent code is subjected to an entropy-based Huffman [30] coding technique to achieve compression. The primary contributions of the proposed scheme are as follows:

- The current framework stands as one of the pioneering endeavours to integrate deep learning methodologies into event data processing. A multilayer neural network called a deep belief network is employed to effectively transform the high-dimensional event data into a low-dimensional latent code. To attain a more compact representation, we utilise entropy-based encoders to perform lossless encoding of the low-dimensional latent features, further compressing the data.
Extensive comparisons were conducted with various lossless benchmark strategies using a diverse standard dataset, encompassing scenes with varying complexity and camera movement. The learning-based framework employed in this study yields a concise latent code for high-dimensional event data, resulting in enhanced compression gains while preserving satisfactory reconstruction quality. Consequently, the proposed framework demonstrates superior performance compared to the benchmark strategies.

2 PROPOSED ARCHITECTURE

DVS generates a high-dimensional multivariate data stream indicating each event’s occurring timestamp, location and polarity. The proposed methodology’s primary goal is to obtain an efficient representation of the high-dimensional input event data. Algorithms such as autoencoders can encode input data into a significantly lower-dimensional representation, encapsulating latent information about the input data distribution. However, autoencoders often need help locating suboptimal local minima, particularly when initialised with large weights [29]. Conversely, initialising with small weights can lead to minute gradients in the early layers, rendering the training of autoencoders with multiple hidden layers infeasible. A deep Belief Network (DBN) is employed to address these issues, which utilizes multiple Restricted Boltzmann Machines (RBMs) [9] to obtain the initial weights for the autoencoder network. DBN is the foundation for the proposed coding framework, facilitating reducing high-dimensional event data into a more compressed form.

The latent version of events is a string of repetitive integer values that are further compressible using an entropy-based encoder such as the Huffman encoder.

2.1 Event Arrangement

Even though the coding framework can encode DVS data on an event-by-event basis, we devise a unique arrangement of event data by applying time aggregation at particular time intervals. The primary benefit of event aggregation is the generation of temporal frames which reduces data size by projecting the DVS spike event stream into a series of frames recording the location histogram count, i.e., the number of event counts at each pixel. We segregate the spike sequence into two separate frames, one for increasing luminance (polarity 1) and one for decreasing luminance (polarity 0). On a particular pixel, if the previous event’s polarity was zero (or one), the next event has a high probability of having zero (or one) polarity due to smooth change in luminance [3]. The polarities of the accumulated spike events have a strong correlation so we record the event count separately for each polarity with each frame having full pixel array resolution. This increases the temporal correlation between frames of the same polarity [14]. We combine the frames with the same timestamp from each polarity into a single super-frame as shown in Figure 3. It comprises a ‘0’ polarity frame on the left and ‘1’ polarity frame on the right resulting in high inter-frame correlation. Hence, super-frames have inherent polarity information and reduce the required frame rate for processing, saving space to store the data.
Figure 4: Workflow of deep auto-encoder with RBM pretraining. (a) Pre-training of RBM weights (b) Unrolling of stacked RBMs (c) Fine-tuning of the unrolled stacked RBMs network.

2.2 Super-frames to Latent Representation

Utilising the inherent spatial, temporal, and statistical correlations present within the event frames of the Dynamic Vision Sensor (DVS) sequence, we propose a compression method that transforms the accumulated spike events into a latent code representation employing a Deep Belief Network (DBN). DBNs are well-suited in autoencoding tasks, particularly for unsupervised feature learning, generative modelling, and handling complex data with multiple abstraction levels. Their proficiency in handling missing data is critical in choosing DBNs, given the prevalence of zero pixels in the super-frames. The DBN architecture consists of a stacked configuration of two-layered stochastic networks comprising visible and hidden layers. Notably, the hidden layer of one Restricted Boltzmann Machine (RBM) is the visible layer for the subsequent RBM in the stack. It is a probabilistic model composed of weights and biases. The structure consists of “n” visible units \( v = (v_1, ..., v_n) \) representing the observed data and “m” hidden units \( h = (h_1, ..., h_m) \) to illustrate the dependencies in the observed data. There is no interconnection among the nodes in each layer to ensure their mutual independence. In binary RBMs, the random variables take the value \( \{0,1\} \in \mathbb{R}^{m+n} \). The hidden and visible variable vectors, \( v \) and \( h \) can be represented by their joint probability density as

\[
p(v, h) = e^{-E(v, h)} \int \int e^{-E(v, h)}
\]

where \( E(v, h) \) is the associated energy function. Since the input data is binary-value, we apply binary-binary energy function [27] as described in (2).

\[
E(v, h) = - \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} v_i h_j - \sum_{j=1}^{n} b_j v_j - \sum_{i=1}^{m} c_i h_i
\]

where, \( w_{ij} \) is the weight associated with the nodes \( v_i \) and \( h_j \). The bias terms for \( j^{th} \) visible and \( i^{th} \) hidden node are \( b_j \) and \( c_i \) respectively. The network assigns a probability to every possible image using the energy function. Adjusting the weights and biases to reduce the energy of a training image, increases the probability of that image. The weights are adjusted as shown in (3).

\[
\Delta w_{ij} = \varepsilon (\langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{recon}})
\]

where \( \varepsilon \) is the learning rate, \( \langle v_i h_j \rangle_{\text{data}} \) and \( \langle v_i h_j \rangle_{\text{recon}} \) are the fraction of times pixel \( i \) and feature \( j \) are on together when driven by data and reconstruction data respectively.

The layer-by-layer learning as shown in algorithm 2 is a very effective way to pre-train the weights of a deep auto-encoder. Each feature layer captures high-order correlations between the events in the layer below. This gradually reveals the low-dimensional, nonlinear structures in a wide range of data sets. The workflow of DBN starts with pretraining process of feature detector layers with initial weights update (3), as shown in Figure 4a. After pretraining is completed, the model is unfolded (Figure 4b) to form a deep auto-encoder network initialised with the pre-trained weights. The training of each RBM maximises the probability of its input data exploiting contrastive divergence (CD) [10] algorithm to update the network parameters. Each feature layer detects the strong and high-order correlations between the units in the layer beneath it. The pre-trained weights of deep auto-encoder are finetuned by replacing
A Novel Approach for Neuromorphic Vision Data Compression based on Deep Belief Network

ICVGIP ’23, December 15–17, 2023, Rupnagar, India

Adding Huffman coding to the DBN framework improves compression gains and peak-signal-to-noise ratio (PSNR). The end-to-end compression ratio of the proposed scheme is reported in Table 1, where we test two different trained models, Model 1 and Model 2, on various datasets at various time aggregation values.

3 EXPERIMENTS
In this work, we first assess the compression gain of the proposed scheme based on two key performance metrics, end-to-end and input-output compression ratios. The end-to-end compression ratio is between uncompressed events and compressed event bitstream size. In contrast, the input-output compression ratio is the input and output frame size ratio. Subsequently, we provide an extensive performance comparison with benchmark strategies regarding compression gains and peak-signal-to-noise ratio (PSNR). The end-to-end compression ratio of the proposed scheme is reported in

Algorithm 1: Deep Belief Encoding of DVS data.

Input: Temporal Resolution $T_e[s]$  
Input: Spike event aggregation duration $T_{agg}[s]$  
Input: Total frame count $F_{total} = \frac{T_e}{T_{agg}}$

1. for $i = 1$ to $F_{total}$ do
2.   for $j = 1$ to $F_{seq}$ do
3.     Record event count with polarity flag "0" in a "M x N" frame.
4.     Record event count with polarity flag "1" in a "M x N" frame.
5.   Merge both event frames to create a super-frame sequence $F_{super}$
6.   Pass $F_{super}$ to fine-tuned model defined in Algorithm 2 to generate its latent space representation $F_l$
7.   $F_l \leftarrow$ Deep Belief Net ($F_{super}$)
8. Apply entropy encoding on $F_l$ to obtain $F_c$
9. $F_c \leftarrow$ Huffman Encoder ($F_l$)
10. Output: Compressed representation of input event frames $F_c$

Algorithm 2: Layer-by-layer training of DBN

Input: DBN, training set $E_{seq}$

1. Initialize network parameters $w, b$
2. $Input \leftarrow X$
3. for all RBM in DBN do
4.   for epoch = 1 to e do
5.     for $k = 1$ to $floor(N_{sample}/N_{batchsize})$ do
6.       $B \leftarrow$ batch from Input
7.       $\Delta w, \Delta b \leftarrow$ Contrastive Divergence
8.       $w \leftarrow w + \gamma \Delta w$
9.       $b \leftarrow b + \gamma \Delta b$
10. $E_{seq} \leftarrow Input \times w + b$
11. Output: Trained Deep Belief Net

3.1 Implementation Details
To test our compression scheme’s performance, we used standard event camera datasets [23]. We choose three different datasets Boxes, Poster and Shapes to evaluate the competence of our proposed scheme at varying degree of scene complexity and camera movement speed. The evaluation was done on a workstation with Intel Xeon 2.30GHz CPU, Nvidia K80 12GB GPU and 12GB RAM specifications. All the experiments were conducted using the Pytorch framework.

We use two different models of the auto-encoder, Model 1 and Model 2. The auto-encoder Model 1 framework comprises an encoder with layers size $(30 \times 30)-1000-500-250-20$ and Model 2 with $(20 \times 20)-600-300-150-20$. The framework possesses a symmetric decoder layer structure. Therefore, a performance comparison of the two models on different datasets with varying scene complexity and motion is depicted in Table 1. The units in the code layer are linear, while other units are logistics. The model’s accuracy increases with lesser units in the code layer, but it has the drawback of longer training time and an increase in low-dimensional code blocks. Hence, we settled with a 20-dimensional code layer maintaining a balance between training time, accuracy and the number of code blocks. The 20-dimensional code blocks are compressed using an entropy-based Huffman coding technique generating a compressed DVS bitstream. The network is trained on 108,000 event blocks derived from the first 10 seconds of the event sequence with 10ms time-aggregation for 20 epochs and 10 batch sizes. The average training time for each sample is approximately 3 hours with a mean squared reconstruction error of 0.164, while the learning rate is set at 0.1. The network is evaluated with test samples obtained with various degrees of time aggregation (0.5ms, 5ms, 10ms, 20ms and 30ms). An average compression bitrate of approximately 13.296 Kbps is achieved, yielding an average PSNR of 64.34 dB. The testing and training sequences are different.

3.2 Experimental Results
As shown in Table 1, we compared our performance with existing benchmark strategies regarding end-to-end compression ratio
on standard datasets with diverse scene complexity and camera movement speed. The framework performs better with the trained Model 1. We outperform all the benchmark coders at almost all time-aggregation values. We also compared our performance with a state-of-the-art event data coder proposed by Khan et al. [12] in terms of end-to-end compression and input-output compression ratio. In both cases of compression metrics, our proposed compression scheme outperforms the TALVEN method [12] at all time-aggregation values as depicted in Figure 5a and Figure 5b respectively. In addition, higher event rate sequences produce better compression performance than low event rate sequences. The

Table 1: Comparison of End-to-End Compression Ratio of Proposed Scheme with Benchmark Coders.

| Tagg | Shapes | Boxes | Poster |
|------|--------|-------|--------|
|      | Motion | Model 1 | Model 2 | Model 1 | Model 2 | Model 1 | Model 2 |
|      | Rotation | 0.58 | 0.24 | 2.82 | 0.64 | 9.20 | 1.94 |
|      | Translation | 0.94 | 0.32 | 9.34 | 3.16 | 14.38 | 6.13 |
| 6 dof | 1.00 | 0.45 | 5.50 | 1.72 | 4.96 | 1.92 |
|      | Average | 0.84 | 0.33 | 5.89 | 1.84 | 9.52 | 3.33 |
|      | Rotation | 6.59 | 2.37 | 27.18 | 6.20 | 95.16 | 21.82 |
|      | Translation | 6.33 | 2.37 | 93.82 | 30.94 | 148.78 | 66.59 |
| 6 dof | 9.30 | 4.16 | 52.95 | 16.38 | 52.15 | 20.40 |
|      | Average | 7.41 | 2.97 | 57.98 | 17.84 | 98.70 | 36.27 |
|      | Rotation | 13.25 | 5.93 | 29.54 | 23.65 | 120.34 | 60.62 |
|      | Translation | 15.46 | 7.69 | 167.21 | 85.05 | 189.90 | 102.55 |
| 6 dof | 19.09 | 8.74 | 60.40 | 57.76 | 76.06 | 31.99 |
|      | Average | 15.94 | 7.46 | 85.72 | 56.15 | 128.83 | 71.72 |
|      | Rotation | 24.56 | 10.27 | 64.15 | 31.93 | 229.33 | 104.74 |
|      | Translation | 35.78 | 15.47 | 279.90 | 121.82 | 353.37 | 162.03 |
| 6 dof | 37.32 | 17.20 | 138.86 | 63.85 | 133.63 | 60.33 |
|      | Average | 32.55 | 14.31 | 160.97 | 72.33 | 238.77 | 109.03 |
|      | Rotation | 37.45 | 18.42 | 92.67 | 46.77 | 309.31 | 155.00 |
|      | Translation | 45.56 | 23.25 | 393.71 | 186.36 | 477.07 | 220.93 |
| 6 dof | 56.14 | 25.22 | 204.00 | 95.03 | 194.89 | 92.68 |
|      | Average | 46.38 | 22.30 | 230.13 | 109.38 | 327.09 | 156.20 |

Figure 5: Comparison of compression performance of Proposed Scheme with respect to TALVEN. (a) Input-Output Compression Ratio (b) End-to-end Compression Ratio (c) PSNR at various Time Aggregation.
Poster sequence, for example, has a very high event rate (4.01 Mega-events/s), while the Shapes sequence has the lowest event rate (0.245 Mega-events/s). The Shapes sequence has low scene complexity and camera speed. Naturally, it should have high compression gains while, in reality, the Poster sequence yields high compression gain. In sequences with high event rate, the delta-coded timestamp need lesser bits to be encoded when compared to low event rate sequences. Additionally, we computed the reconstruction quality of our proposed model with the help of the peak-signal-to-noise ratio (PSNR) metric. The proposed model can reconstruct the dynamic vision sensor event data from the compressed bitstream with a high PSNR value, as shown in Figure 5c.

4 CONCLUSION

The dynamic vision sensor effectively captures changes in per-pixel intensity, generating an asynchronous stream of event data. Our research introduces a novel event data arrangement and compression framework, leveraging the Deep Belief Network (DBN). Notably, our framework represents one of the pioneering efforts to employ deep learning in event data processing. Time aggregation of spike events emerged as a valuable strategy, enhancing efficiency by contextualizing specific event groups. Leveraging the super-frame arrangement, we observed heightened spatial and temporal correlations among events, effectively utilized by the DBN to extract latent feature codes. The extensive performance comparison study presented in this paper substantiates the superiority of our proposed coding framework over benchmark compression schemes. These findings highlight the potential of deep learning-based approaches in significantly improving event data compression efficiency. In conclusion, our research contributes valuable insights and advancements to the field of event data processing for dynamic vision sensors.

REFERENCES

[1] Jyrki Alakuijala and Zoltan Szabadal. 2016. Brots compressed data format. Technical Report.
[2] Hafeez Ullah Amin, Mohd Zuki Yusoff, and Rana Fayyaz Ahmad. 2020. A novel approach based on wavelet analysis and arithmetic coding for automated detection and diagnosis of epileptic seizure in EEG signals using machine learning techniques. Biomedical Signal Processing and Control 56 (2020), 101707.
[3] Zhichao Bi, Sewei Dong, Yonghong Tian, and Tiezun Huang. 2018. Spiker coding for dynamic vision sensors. In 2018 Data Compression Conference. IEEE, 117–126.
[4] Davis Blalock, Samuel Madden, and John Gutttag. 2018. Sprintz: Time series compression for the internet of things. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2, 3 (2018), 1–23.
[5] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liou, and Tobi Delbruck. 2014. A 240×180 130 dB 3μs latency global shutter spatiotemporal vision sensor. IEEE Journal of Solid-State Circuits 49, 10 (2014), 2333–2341.
[6] Vincent Chan, Shih-Chii Liu, and Andr van Schalk. 2007. AER EAR: A matched silicon cochlea pair with address event representation interface. IEEE TCAS-I: 54, 1 (2007), 48–59.
[7] Yann Collet and Murray Kucherawy. 2015. Zstd Compression and the application/zstd Media Type. Technical Report.
[8] Peter Deutsch and Jean-Loup Gailly. 1996. Zlib compressed data format specification version 3.3. Technical Report.
[9] Geoffrey Hinton. 2010. A practical guide to training restricted Boltzmann machines. Momentum 9, 1 (2010), 926.
[10] Geoffrey Hinton. 2002. Training products of experts by minimizing contrastive divergence. Neural computation 14, 8 (2002), 1771–1800.
[11] Najib Khan, Khurram Iqbal, and Maria G Martini. 2020. Time-Aggregation-Based Lossless Video Encoding for Neuromorphic Vision Sensor Data. IEEE Internet of Things Journal 8, 3 (2020), 596–609.
[12] Hamme Kim, Ankur Hunda, Ryad Benosman, Sio-Hoi Ieng, and Andrew J Davison. 2008. Simultaneous mosaicing and tracking with an event camera. J. Solid State Circ 43 (2008), 566–576.
[13] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E Shi, and Ryad B Benosman. 2016. Hots: a hierarchy of event-based time-surfaces for pattern recognition. IEEE TPAMI 39, 7 (2016), 1346–1359.
[14] Daniel Lemire and Leonid Boytsov. 2015. Decoding billions of integers per second through vectorization. Software: Practice and Experience 45, 1 (2015), 1–29.
[15] Bing Li, Lin Zhang, Zhaohui Huang, and Quan Dong. 2014. Implementation of LZMA compression algorithm on FPGA. Electronics Letters 50, 21 (2014), 1522–1524.
[16] Patrik Lichtsteiner, Christoph Posch, and Tobi Delbruck. 2008. A 128 × 128 120 dB 15μs latency asynchronous temporal contrast vision sensor. IEEE journal of solid-state circuits 43, 2 (2008), 556–576.
[17] Wensiang Liu, Faqiang Mei, Chenghui Wang, Maire O’Neill, and Earl E Swartzlander. 2018. Data compression device based on modified LZA algorithm. IEEE Transactions on Consumer Electronics 64, 1 (2018), 110–117.
[18] Misha Mahowald. 1994. The silicon retina. In An Analog VLSI System for Stereoscopic Vision. Springer, 4–65.
[19] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso Garcia, and Davide Scaramuzza. 2018. Event-based vision meets deep learning on steering prediction for self-driving cars. In Proceedings of the IEEE/CVF CVPR. 5419–5427.
[20] Abdel-rahman Mohamed, George Dahl, Geoffrey Hinton, et al. 2009. Deep belief networks for phone recognition. In Nips workshop on deep learning for speech recognition and related applications, Vol. 1. Vancouver, Canada, 39.
[21] Elias Mueggler, Basil Huber, and Davide Scaramuzza. 2014. Event-based, 6-DOF pose tracking for high-speed maneuvers. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2761–2768.
[22] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. 2017. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. The International Journal of Robotics Research 36, 2 (2017), 142–149.
[23] Christoph Posch, Daniel Matolin, Rainer Wohlgemuth, Michael Hofstätter, Peter Schön, Martin Litzenberger, Daniel Bauer, and Heinrich Garn. 2010. Live demonstration: ATIS camera with full-custom aec processor. In Proceedings of IEEE ISCAS. IEEE, 1392–1392.
[24] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. 2019. Events-to-video: Bringing modern computer vision to event cameras. In Proceedings of the IEEE/CVF CVPR. 3857–3866.
[25] Amin Rigi, Fariborz Baghaei Naeni, Dimitrios Makris, and Yahya Zweiri. 2018. A novel event-based incipient slip detection using dynamic active-pixel vision sensor (DAVIS). Sensors 18, 2 (2018), 333.
[26] Chang Shao, Fan Yang, Dexian Huang, and Wenzhang Lyu. 2014. Data-driven soft sensor development based on deep learning technique. Journal of Process Control 24, 3 (2014), 223–233.
[27] Antonino Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. 2018. Ultimate SLAM! Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios. IEEE Robotics and Automation Letters 3, 2 (2018), 994–1001.
[28] Jing Wang, Haibo He, and Danil V Prokhorov. 2012. A folded neural network autoencoder for dimensionality reduction. Proceedia Computer Science 13 (2012), 120–127.
[29] Lianming Zhang, Fengyong Li, and Chuan Qin. 2022. Efficient reversible data hiding in encrypted binary image with Huffman encoding and weight prediction. Multimedia Tools and Applications 81, 20 (2022), 29347–29365.
[30] Alex Zihao Zhu, Liangzheng Yuan, Kenneth Chaney, and Kostas Daniilidis. 2019. A novel approach for neuromorphic vision data compression based on Deep Belief Network.

ICVGIP ’23, December 15–17, 2023, Rupnagar, India