Evaluation of the impact of lossy compression on event camera-based computer vision tasks

Bowen Huang[a], Davi Lazzarotto[a], and Touradj Ebrahimi[a]

[a]Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

ABSTRACT

In the field of image acquisition, Dynamic Vision Sensors (DVS) present an innovative methodology, capturing only the variations in pixel brightness instead of absolute values and thereby revealing unique features. Given that the primary deployment of DVS is within embedded systems characterized by their constrained transmission and storage capabilities, the emphasis on data compression becomes significant. Nonetheless, such a compression could potentially compromise the efficacy of computer vision (CV) applications. This study investigates the implications of a lossy compression technique, premised on point cloud representation, for event data in CV tasks. Multiple scenarios under various compression intensities are applied to event data, and the experiments indicate the feasibility of attaining reduced bitrates while incurring minimal impact in CV task performance.

Keywords: Event-based vision, point cloud representation, lossy data compression

1. INTRODUCTION

The world has observed in the past decades an explosion in the amount of media being produced, transmitted, and stored. Frame-based cameras are, among many other technological advancements, responsible for this phenomenon, and have been used to acquire vast amounts of dynamic visual information. The produced digital data is usually represented as a succession of two-dimensional arrays of pixels, with a sampling rate high enough to fool the human vision system into perceiving it as continuously change over time. This frame rate is constant and specified by an external clock, typically at around 30 frames per second or more (fps).

While such cameras are adequate for many scenarios, some applications where the brightness of the scene can change faster than the Nyquist rate require shorter response times, and increasing the frame rate indefinitely would result in impractical amounts of data. Event cameras, also known as Dynamic Vision Sensors (DVS), are interesting alternatives in these cases. Instead of recording the entire field of view at a fixed frequency, DVS records changes in the measured intensity of the brightness over independent individual pixels asynchronously. This biologically-inspired approach makes DVS capable of achieving very high temporal resolutions and dynamic range, as well as low latency and reduced power consumption. These novel characteristics bring new possibilities for all kinds of applications and various research activities have been conducted to apply DVS to classic computer vision (CV) tasks, such as recognition, segmentation, and reconstruction.

Data generated by DVS are represented as events that contain the location and timestamp of brightness changes. Each event is triggered when the brightness of a pixel either increases or decreases by a value equal to the contrast sensitivity C. In the Address Event Representation (AER) protocol, the event contains the coordinates (x, y) of the pixel, the timestamp t when the event occurs, and the polarity p of the lightness variation, which is set either to 1 or -1 and refers to the direction of the brightness variation (C or $-C$, respectively). Considering the different requirements on the resolution of raw event data, 96 or 64 bits are used to represent the tuple (x, y, t, p), where 1 bit is assigned to indicate polarity, 64 or 32 bits are assigned to timestamp and the rest used to signal spatial coordinates. Since they do not use synchronous clocks for data transmission, event cameras can record events, usually with a precision of microseconds. When the spatial resolution is also increased, the amount of output data can also be a burden for transmission and storage, especially for embedded systems with limited resources. Therefore, finding efficient compression methods for event data streams is important to several DVS

Further author information: (Send correspondence to the authors)
E-mail: bowen.huang@epfl.ch, davi.nachtigalllazzarotto@epfl.ch, touradj.ebrahimi@epfl.ch

applications. While general-purpose entropy coding and integer coding compression algorithms can be utilized, methods specifically developed for event data compression have demonstrated increased performance.

Several approaches proposed in the literature conduct lossless compression of event data, allowing for complete recovery of the information at the decoder side. However, the resulting compression ratio of such techniques is low and does not meet the requirements of some applications, which have to rely on lossy approaches in order to enable higher compression. Among the specifically designed methods, time aggregation is one of the most popular strategies, where events during a fixed time interval are combined into one event frame (EF) and conventional image coding is applied to compress them. However, the time interval is fixed for EFs, and thus such methods cannot maintain the native time resolution of event cameras. Recent works[1,2] resorted to the representation of event data as point clouds, casting the time values as a spatial dimension and denoting each event by coordinates (x, y, t), which are then encoded with lossless or lossy point cloud compression methods.

The impact of compression methods on the quality of regular images is usually measured by objective quality metrics or by subjective experiments. However, since event data is not made for direct visualization for humans, it cannot be evaluated neither subjectively nor objectively through image-based metrics such as PSNR or SSIM.[3] One meaningful alternative that has not been explored in the literature is to assess the impact of lossy compression on the performance of state-of-the-art CV algorithms, which are usually applied only to the original and therefore undistorted event data.

This paper presents an evaluation study of the impact of a lossy event data compression method on the performance of four distinct vision tasks representative of several important applications: recognition, video reconstruction, optical flow estimation, and depth estimation. A dataset containing event data sequences is selected, represented as point clouds following the pipeline proposed in,[1] and compressed with G-PCC[4] using lossy configurations. The decoded sequences are then fed to state-of-the-art algorithms for the selected vision tasks. Their performance is evaluated using different metrics and compared to the case where uncompressed data is used as input, allowing to draw conclusions on the impact of such compression methods in practical applications.

2. RELATED WORK

2.1 Event based Vision

The paradigm shift brought by event data when compared to dense images has opened new perspectives in many applications relying on computer vision tasks, especially those requiring higher speed and lower latency. However, the sparse nature of such data calls for the adaptation of image-based algorithms to handle brightness variations rather than absolute pixel values. For example, object and shape recognition are challenging tasks in event cameras. Most current datasets for frame-based images are acquired with static camera and scene, a condition where DVSs are not likely to perform well as events are generated by moving edges. In contrast, they would probably excel in applications with moving objects or cameras where motion blur would affect the performance of traditional frame-based cameras. However, such evaluation conditions have not yet been adequately defined, and the lack of large-scale annotated datasets remains a challenge.

Even so, research in the field has been improving the performance of such algorithms. Early works performed template matching for the detection of known simple shapes.[5,6] Other researchers[7] have relied on a similar approach to detect low-level features, which are then used to recognize more complex shapes with a classifier. Learning-based methods can obtain images from the events prior to recognition[8] or convert a trained network into a spiking neural networks (SNN) that can take event data as input.[9] In this work, we evaluate a method[10] that converts an event stream into a set of points lying on a three-dimensional grid using differentiable operations, which is then used in combination with recognition methods based on convolutional networks and trained end-to-end.

Another example is video reconstruction, which is a task that, in an ideal noise-free scenario, can be easily computed by integrating the events over time, requiring the initial offset image containing absolute values. However, past work has demonstrated that it is possible to estimate brightness from input events even without an initial condition.[11] Early works[12] reconstructed images with events obtained by a rotating camera on static scenes. In more recent efforts,[13] regularization has been added, enabling the model to operate in any type of camera movement and scenes containing dynamic objects by performing both brightness and optical flow estimation. Recent learning-based approaches[14] replaced hand-crafted regularizers with learned functions obtaining better image quality. Studies[15] have also demonstrated that the images generated by such methods can become input to CV algorithms such as objective classification or visual odometry and achieve better performance when compared to directly using event data. This method[15] is taken as the basis for evaluation in this paper.

Event cameras are naturally suitable for motion analysis tasks, such as optical flow estimation, in which the velocity of objects in the image plane must be computed. If with conventional images, the flow can be derived from solving sets of equations obtained through the spatial and temporal derivatives over two consecutive frames, event data contain neither absolute brightness nor spatially continuous information across all pixels in a given region of space. However, DVS is very attractive for this task since it provides edge information in a more direct way, crucial for a correct estimation of optical flow. Early works[16] attempted a direct adaptation of image-based algorithms for event data, but the small number of events generated when an edge crosses a pixel hinders the estimation of spatial and temporal derivatives of the brightness, resulting in inconclusive results. The work from[17] takes into account the geometrical distribution of events and considers that the movement of an edge can be represented by a surface in the xyt volume. The optical flow can then be extracted from a plane fitted to the event data in this three-dimensional space. Recent methods have leveraged the vast amount of data available in current datasets to develop algorithms based on deep learning, which are trained with event data either supervised[18] by image data captured by a DAVIS camera or in an unsupervised manner.[19,20] A recent approach[21] has proposed an image-event fusion model allowing the prediction of optical flow both dense in space and continuous in time and is used for the evaluation conducted in this paper.

In addition, event data can be used for 3D vision tasks, such as depth estimation and simultaneous localization and mapping (SLAM). The problem of depth estimation using DVS has been mainly addressed by two categories of methods: instantaneous stereo or monocular. The majority of the works in the literature focus on the first class, where two event cameras are placed at a fixed known distance between each other and share the same clock, meaning that the timestamp of one event captured by both cameras is related to the same reference. These works usually follow a two-step approach: the events from the two cameras are matched, and then the spatial position of the point is obtained through triangulation. The first step is usually the most computationally heavy and therefore different algorithms have been suggested to accomplish this task. Local methods[22,23] compare the neighborhoods of different events in both cameras to determine if they correspond to the same spatial location. However, they are prone to ambiguities, and global methods that consider additional constraints tend to produce better results. In this category, there are extensions[24,25] of the cooperative stereo algorithm[26] and methods based on belief propagation over Markov Random Fields.[27,28] Recent works have also proposed to integrate data from LiDAR sensors, which already measure depth at sparsely distributed points. The event data is in this case used to enhance the results obtained with LiDAR. The work presented in[29] uses both convolutional and recurrent neural networks for this integration, predicting much denser depth maps, and is adopted for evaluation in this paper.

2.2 Event Data Compression

In the first lossless compression strategy for event data,[30] the authors designed two different modes termed Address-Prior (AP) and Time-Prior (TP) to deal with the spatially decentralized data stream and spatially centralized data stream separately. For each macro cube formed by accumulated event data of a certain time interval, the two modes are applied synchronously and the one achieving the better compression performance is selected. A CABAC entropy encoder is cascaded to generate the final bitstream.

On the other hand, aggregating event data of fixed time intervals into event frames (EFs) is meaningful for high-level tasks, and conventional video coding can be applied to these EFs. In an effort to leverage the performance of popular video compression standards,[31] a method was proposed where the pixels of produced EFs record the number of event data during a time interval, producing two different EFs concurrently according to their polarity. They are then concatenated into one superframe and the sequence of superframes is encoded by frame-based video coding, such as HEVC. Schiopu *et. al.*[32] also proposed using EFs to compress DVS data. A more efficient data structure of up to eight EFs is introduced and a binary map signaling the positions where at least one event occurs in the EFs is used to relieve the sparsity of the DVS data stream. Besides, the number of events for each signaled position and their EF index are also encoded together by conventional video coding. Another compression method was introduced[33] where, at the encoding stage, events are first aggregated over time to form polarity-based event histograms. A quadtree (QT) segmentation map derived from the adjacent intensity images is then used to solve the lack of spatial structure of event data. The histograms are variably sampled via Poisson Disk Sampling and the entropy coding strategy is cascaded to further improve the compression ratio.

Though specifically designed for the DVS data stream, the above-mentioned methods are based on aggregated EFs, leading to a loss of time resolution. If the event data is aggregated via a fixed time interval, it is difficult to satisfy different computer vision tasks with different time resolution requirements. Besides, the sparsity of DVS data makes conventional video coding designed for video frames of dense pixels inefficient. Recent works[1,2] proposed to regard event data sequences as a sequential sparse point cloud with coordinates of (x, y, t) and utilize point cloud encoding such as G-PCC[4] to compress the event point cloud. Since the sparsity of the event data is similar to point clouds, which suit the characteristics of the point cloud coding, time and space redundancy can be significantly reduced leading to high compression ratios.

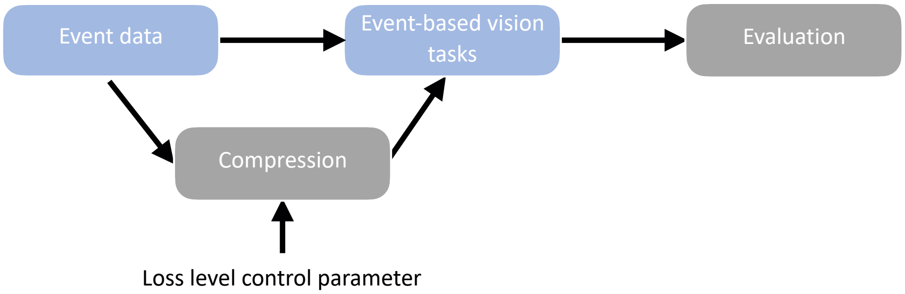

Figure 1. The proposed assessment pipeline.

3. PROPOSED ASSESSMENT PIPELINE

To assess the effect of lossy compression on event data, the event stream is restructured into a 3D point cloud representation, where the timestamp t serves as the third axis. Subsequently, point cloud coding techniques are applied to the resulting event point cloud at varying levels of lossy compression. The decompressed versions are then reverted back to their original format for subsequent testing. The assessment pipeline is depicted in Fig.1. This section expounds on the background of point cloud compression (PCC) and the critical stages within the proposed assessment pipeline.

3.1 Point Cloud Compression

A point cloud is a data structure that comprises numerous distinct points, with each point containing coordinates, known as geometry information, and additional features such as color or reflectance, referred to as attributes. In an effort to enhance storage and transmission efficiency, point cloud compression techniques have been proposed to exploit the inherent redundancy within 3D space. Two distinct approaches exist for point cloud compression: one directly compresses the 3D data structure, while the other projects 3D data onto 2D planes and subsequently employs conventional 2D video coding.

The Moving Picture Expert Group (MPEG) has been involved in the standardization of Point Cloud Compression (PCC) since 2018, developing two coding approaches tailored to distinct point cloud application scenarios.[34] One approach, Video-based PCC (V-PCC), is specifically designed for dynamic point clouds, while the other, Geometry-based PCC (G-PCC), is intended for static and dynamically captured point clouds, such as those derived from LiDAR sensors. V-PCC relies on 3D-to-2D projections, better suited for dense point clouds where points are primarily situated on the surface of the 3D object. Indeed, point clouds with a significant number of inner points may lead to overlapping projections on 2D frames, resulting in a reduction of reconstruction quality.

Contrasting with V-PCC, G-PCC employs an octree structure to partition the point cloud into recursive cubes. G-PCC is better suited to irregular and sparser point clouds, mirroring the sparsity inherent in event data. Furthermore, the octree structure in G-PCC enables it to encode all points of the 3D object directly, thereby enhancing its utility for a broader range of data that resemble point cloud structures.

Draco serves as another prevalent codec for 3D object compression. It compresses point clouds by transforming them into mesh structures and subsequently performs mesh compression. Draco's compression speed notably outpaces that of both V-PCC and G-PCC. However, since it was originally designed for mesh compression, managing the distortion in Draco is more challenging when compared to its counterparts.

With the advancements in machine learning and deep learning techniques, neural networks have also been used in the field of PCC. Numerous studies have focused on leveraging Deep Neural Network (DNN)-based PCC strategies. A significant proportion of these DNN-based methods employ the architecture of an auto-encoder, incorporating residual blocks and attention mechanisms to enhance compression performance. Importantly, some of these DNN-based methods can process the point cloud in its raw data structure, eliminating the necessity for voxelization. This attribute of DNN-based techniques effectively prevents the loss usually introduced by coordinate quantization.

From the findings of our preliminary experiments,[1] G-PCC was selected as the point cloud coding in this paper. This decision was influenced by Draco's observed suboptimal compression rates and its challenging controllability. Additionally, compared to handcrafted models, DNN-based approaches manifested reduced generalization capabilities.

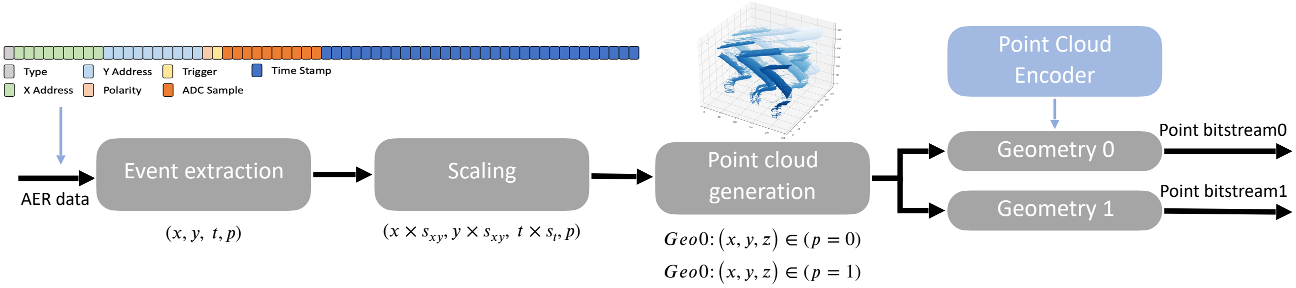

Figure 2. The framework for event data compression method based on point cloud representation. In this framework, two point clouds are generated based on polarity and each compressed separately.

3.2 Coding Procedure

The overall methodology for event compression is illustrated in Fig.2. The initial stage involves the construction of a tridimensional point cloud representation for the event data by considering the timestamp as a third dimension. It has been suggested by[35] that event data of identical polarity exhibit more robust spatial and temporal correlations. This is due to the fact that points belonging to a single object are likely to exhibit synchronous movements, thus resulting in events of a consistent polarity in space, which subsequently persist over time. As a result, events are segregated according to their distinct polarities - positive or negative - and two individual event point clouds are generated.

To keep the consistence of the values of timestamp comparable to spatial coordinates in their order of magnitude, the timestamp of each event point is multiplied with a scaling factor, and the event point cloud is centralized. To fragment the continuous event data stream into discrete point cloud frames, events are accumulated to a fixed quantity, thus yielding an event point cloud comprising a constant number of points.

The parameters mentioned above are determined based on experimental findings.[1] The encoding procedure for the proposed method can be summarized as follows. The incoming event data stream is accumulated up to a predetermined threshold, thus creating a three-dimensional point cloud with coordinates (x, y, t). This is followed by scaling and centralization operations to ensure that the coordinate range of the generated event point cloud is suitable for the subsequent point cloud encoding. Once the event point cloud is generated, a point cloud encoder is employed to compress the event point clouds at varying loss levels. This is achieved by adjusting different quantization scale parameters.

The decoding procedure is essentially the inverse of the encoding process. During this stage, the decompressed bitstream is reconverted into its original format, albeit with some information loss as a consequence of the lossy compression. An advantage of such a flexible coding approach is that the decompressed event point clouds can be re-aggregated at any desired temporal resolution to facilitate subsequent CV tasks.

4. EVENT-BASED CV TASKS

In order to assess the effect of lossy compression on event-based vision tasks, a diverse range of CV operations are selected. These operations encompass various categories, including recognition and video reconstruction tasks, which represent information comprehension and reconstruction, respectively. Given the focus on motion analysis and 3D-related applications, optical flow estimation and depth estimation are also selected. In this section, the methodologies employed in the selected CV tasks are briefly outlined, with a particular emphasis on the format of the input data and the objective metrics used. The goal is to provide a comprehensive understanding of the experimental results.

4.1 Recognition

As a classic CV task, recognition is a fundamental application of visual information understanding. Object recognition based on event data holds distinct advantages such as superior dynamic range, reduced latency, and minimal motion blur.

In the selected recognition algorithm,[10] the authors propose a universal framework to convert asynchronous event streams into grid-based representations, and the conversion is performed through differentiable operators, which allows end-to-end optimization for recognition. To ensure differentiability, the events are defined as the *Event Measurement Field* regulated by a time-stamp measurement $f_{\pm}(x, y, t) = \frac{t-t_0}{\Delta t}$, and a suitable aggregation kernel is used to derive a meaningful signal from the event spikes. After kernel convolutions, the convolved signal is sampled according to a predefined voxel grid to generate a grid representation called *event spike tensor (EST)*. The entire process can be described as follows:

$$S_{\pm}[x_l, y_m, t_n] = (k * S_{\pm})(x_l, y_m, t_n) = \Sigma_{e_k \in \varepsilon_{\pm}} f_{\pm}(x_k, y_k, t_k) k(x_l - x_k, y_m - y_k, t_n - t_k) \tag{1}$$

where the x_l, y_m, t_n lie on the voxel grid $\{0 \sim W-1, 0 \sim H-1, t_0 \sim t_0 + B\Delta t\}$, and $k(x, y, t)$ is the convolution kernel. Commonly, k is handcrafted functions, such as alpha-kernel or exponential kernel, and the authors propose to use multilayer perception (MLP) to estimate the activation map for each grid location in the representation for better performance.

In terms of object recognition, the researchers employed a ResNet-34 architecture,[36] modifying the first and last layers of the pre-trained model with randomly initialized weights to accommodate discrepancies in the number of input channels and output classes between the pre-trained model and the proposed method. To evaluate the performance, recognition accuracy is used, which is defined as $(TP + TN)/ALL$, where TP is true positive, TN is true negative, and ALL is the total number of samples.

4.2 Video Reconstruction

Video reconstruction takes event data stream as input and outputs synthesized content. In our experiments, E2VID[15] was selected, a recurrent convolutional neural network developed specifically for event-based video reconstruction. To accommodate the event data within a neural network framework, the authors used an event tensor representation predicated on a spatiotemporal voxel grid, formulated as follows:

$$\mathbf{E}(x_i, y_i, p_i, b) = \Sigma_i^N max(0, 1 - |b - t_i^*|) \tag{2}$$

where $t_i^* = \frac{B-1}{t_{end} - t_{start}}(t_i - t_{start})$. In the proposed implementation, the temporal bin B is set to 5. By using the recurrent layer, E2VID only takes the event data stream as input, without requiring the last reconstructed images as explicit memory. After generating the reconstruction result, the frames are rescaled using robust min/max normalization to obtain the final output.

In the presence of ground truth frames, the PSNR (Peak Signal-to-Noise Ratio) is commonly employed as an objective metric for video reconstruction. For each ground-truth frame, the reconstructed image with the closest timestamp is selected. To ensure a fair comparison, local histogram equalization is implemented on each ground-truth frame and reconstructed frame prior to computing PSNR. This procedure ensures that the intensity values lie within the same range across frames.

4.3 Optical Flow Estimation

Given the high dynamic range and exceptional temporal resolution of event cameras, they naturally excel in dynamic analysis in complex environments, such as optical flow estimation. Considering that event streams reveal only brightness changes rather than absolute brightness, it is challenging to identify the spatial photometric consistency among sparse pixels and consequently estimate dense optical flow. To address this issue, the selected algorithm DCEIFlow[21] utilizes a single image as an anchor, and the event streams are input into the neural network to yield continuous optical flow estimations.

The event stream is aggregated into a 3D event volume in the DCEIFlow network. The event representation can be described as follows:

$$\mathbf{E}(x_i, y_i, p_i, b) = \Sigma_i^N max(0, 1 - |b - \frac{B-1}{t_{end} - t_{start}}(t_i - t_{start})|) \tag{3}$$

where B is the temporal bin and is set to 5. t_{start}, t_{end} are the start and end timestamps of the event streams, respectively. The two polarities are concatenated to generate the final event volume.

A commonly used metric for optical flow evaluation is the average End Point Error (EPE):

$$EPE = \frac{1}{m} \cdot \Sigma_m \sqrt{(F_x^{pred} - F_x^{gt})^2 + (F_y^{pred} - F_y^{gt})^2} \tag{4}$$

where x and y are the horizontal and vertical directions. Besides, following KITTI,[37] the outlier metric is also used, which reports the percentage of points with endpoint errors greater than 3 pixels and 5% of the magnitude.

4.4 Depth Estimation

In addition to 2D CV tasks, event-based 3D vision tasks also garner significant interest, particularly as the integration of event data with other data modalities in 3D-related tasks holds substantial practical implications, such as depth estimation.

LiDAR sensors are frequently used for depth estimation. However, the 3D information gathered by LiDAR is inherently sparse, thereby presenting a limiting factor. The selected algorithm, ALED,[29] concentrates on integrating LiDAR and event data, utilizing the events to enrich the sparse LiDAR data. In this approach, each event is assigned two depth estimations, one prior to the occurrence of the event and one afterward. ALED network is a fully convolutional recurrent network, designed to process both LiDAR and event data concurrently. The event data $\{e_i = (x_i, y_i, p_i, t_i)\}_{i=1}^N$ is represented as *Discretized Event Volumes*, which is shown in Eq.2. Following the setting of the original paper, B is set to 5, and the negative and positive polarity bins are concatenated along the first dimension.

The objective assessment of depth estimation is conducted at various cut-off distances, ranging from 10m to 200m. At each distance, the average absolute error is calculated. Given the ground truth depth map D_{gt} and the estimated depth map D_{pred}, the average absolute error is defined as follows:

$$AAE = mean(|D_{gt} - D_{pred}|) \tag{5}$$

Since two depth estimations are given to each event in the paper, the final AAE is calculated by averaging the AAE of depth estimation before and after each event. In addition, events are classified to different objects based on the estimated depth, and the percentage of correctly classified events is also reported.

5. EVALUATION CONDITIONS

5.1 Dataset

For a fair comparison, the testing datasets for each task are selected in accordance with the original paper. This includes the MVSEC dataset[38] for optical flow estimation and depth estimation, the N-Caltech101 dataset provided by[10] for recognition, and the DAVIS dataset[39] for video reconstruction. Given time constraints, only portions of the MVSEC and DAVIS datasets were used. However, the selected sequences incorporate a variety of scenarios, including both outdoor and indoor environments, to ensure a comprehensive evaluation. The details of the selected sequences are summarized in Tab.1.

Task	**Dataset**	**Notes**
Recognition	N-Caltech101	testing dataset in N-Caltech101, containing 101 sequences
Optical flow estimation	MVSEC	indoor-flying1/2 and outdoor-days1
Depth estimation	MVSEC	indoor-flying1/2 and outdoor-days1
Video reconstruction	DAVIS	boxes 6dof, hdr poster, office zigzag, outdoors walking, poster translation and slider depth

Table 1. Detailed dataset information for the selected tasks.

5.2 Experiment Settings

In the subsequent experiments, event point clouds are generated according to a fixed number of points, with the number of points per frame controlled at around 1'000'000. The temporal scaling factor is set to 1×10^6 to minimize information loss during pre-processing, and to keep the consistence of the order of magnitude, (x, y) is multiplied by a spatial scaling factor 1×10^3.

The degree of lossy compression for G-PCC is modulated by the *position quantization scale* parameter, which is set to values 1, 1/5, 1/50, 1/100, and 1/500 during the experiments. By definition, a *position quantization scale* of 1 corresponds to lossless compression, while a value less than 1 indicates lossy compression. For each task, the performance of the selected algorithm was first assessed according to its corresponding metric using uncompressed event data as the input. The event data decompressed at the selected levels were then served as input and the performance was again evaluated. The percentual difference between these two values was finally obtained, denoting the impact of compression in the performance of the method.

The recognition algorithm was trained as per the instructions using the N-Caltech101 dataset, while all other tasks utilized pre-trained checkpoints provided. It should be noted that the output frames from the video reconstruction algorithm displayed notable brightness discrepancies compared to the ground truth images in the DAVIS dataset. Therefore, a histogram-based brightness calibration function from the *scikit-image* library was used prior to metric calculation.

6. RESULTS AND DISCUSSION

The performance variations associated with G-PCC compression of event data are illustrated in Fig.3. Broadly, event-based vision tasks exhibit minimal sensitivity to lossy compression. However, the impact of compression can be more perceived at higher compression ratios, and the extent of this influence diverges across distinct tasks. Such variability is rooted in the disparities in input data and the different requirements on the temporal resolution for different applications.

As observed in Fig.3, across all tasks, the performance degradation is minor for the first four loss levels, *i.e.*, the lossless compression and the lossy compression with *position quantization scale* of $1/5, 1/50, 1/100$. This can be partially attributed to the composition of the input data. For depth estimation and optical flow, the impact of information loss in event data may be alleviated by the inclusion of other input data modalities, such as LiDAR data and images. In particular, the depth estimation method displays higher robustness to compression as the LiDAR data serves as a continuous input.

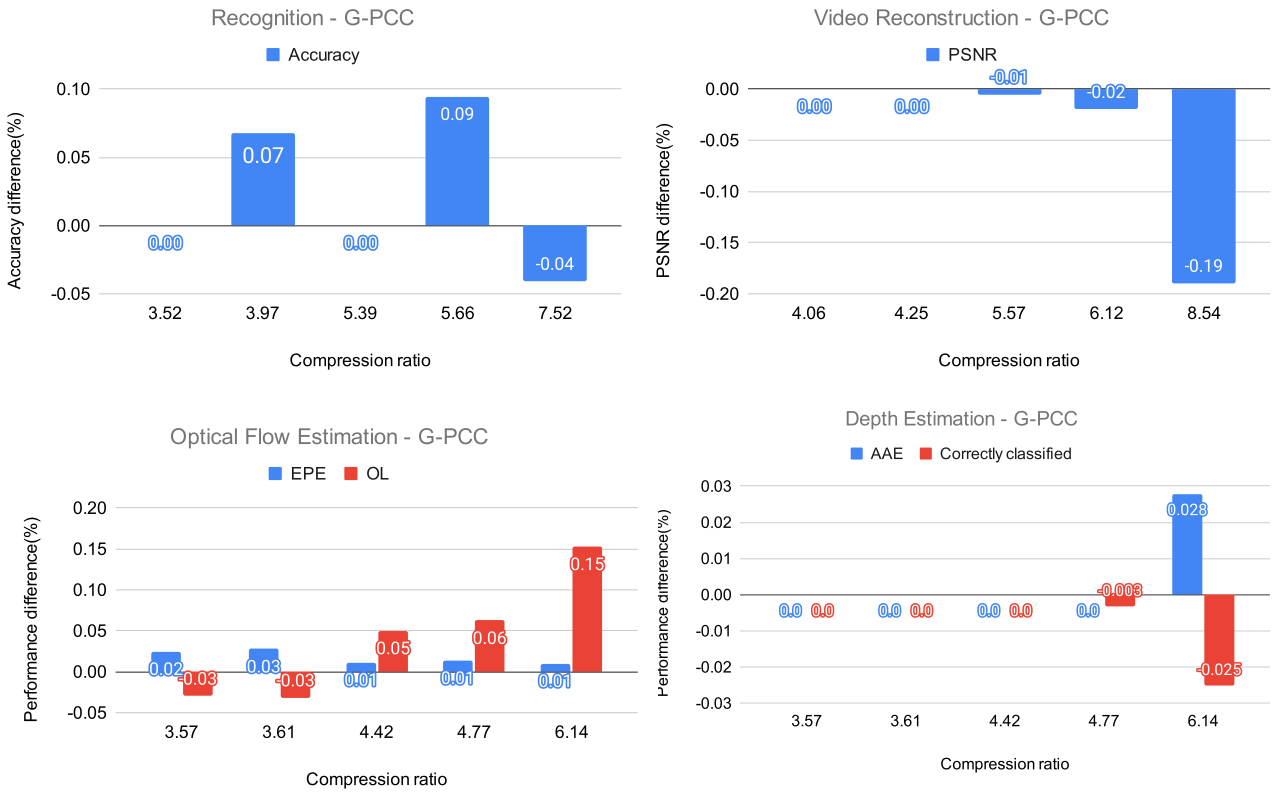

Figure 3. Results of relative performance differences in selected event-based vision tasks using G-PCC compressed event data.

In the case of recognition and video reconstruction, where the sole input is event data, the omission of some input details does not significantly hinder the overall information and subsequent recognition accuracy, since the process is based on the accumulation of event data, whose resolution is decided by the temporal bins in the representation. Moreover, for video reconstruction, the reconstructed frames are compared with ground truth frames derived from a traditional camera operating at a comparatively lower frame rate of approximately 20 fps. As such, the reconstruction results maintain a comparable level of quality. Nevertheless, tasks related to visual information reconstruction demand a greater level of details when compared to tasks focused on understanding. Evidently, the performance degradation in video reconstruction escalates more obviously than in recognition. Interestingly, certain tasks such as recognition even demonstrate enhanced results with lossy compression. The possible reason is that the process of lossy compression may eliminate some redundant information, which can simplify the subsequent CV tasks and enable easier capture of important information. However, a more detailed explanation warrants further investigation.

With escalating loss levels, there's a notable uptick in the compression ratio, paralleled by a marked and swift degradation in performance, exemplified in tasks like video reconstruction and optical flow estimation. The optical flow estimation algorithm predominantly employs a singular input image as its optical flow estimation anchor. Such dependency renders optical flow estimation particularly susceptible to performance decrements when event streams lack precise information, which is evidenced by the elevated outlier metric. The performance of video reconstruction is also linked to the level of details in the event data, especially the high-frequency information, which is also the central target during lossy compression optimization. However, the evaluated levels of compression were not enough to produce a significant effect on the subject visual quality of the output even for the highest compression ratio, as showcased by Fig.4.

As a preliminary conclusion, it can be summarized that the performance loss is contingent upon both the event representation and the input data format. Notably, for CV tasks that integrate event data with other sensor modalities, such as LiDAR and images, the impact of lossy compression remains constrained, rendering the algorithms intrinsically resilient. For content reconstruction and generation tasks that rely purely on event data, lossy compression should be applied more carefully to avoid degradation in quality. In future works, extensive evaluation will be conducted on diverse event vision-based tasks, using various PCC codecs.

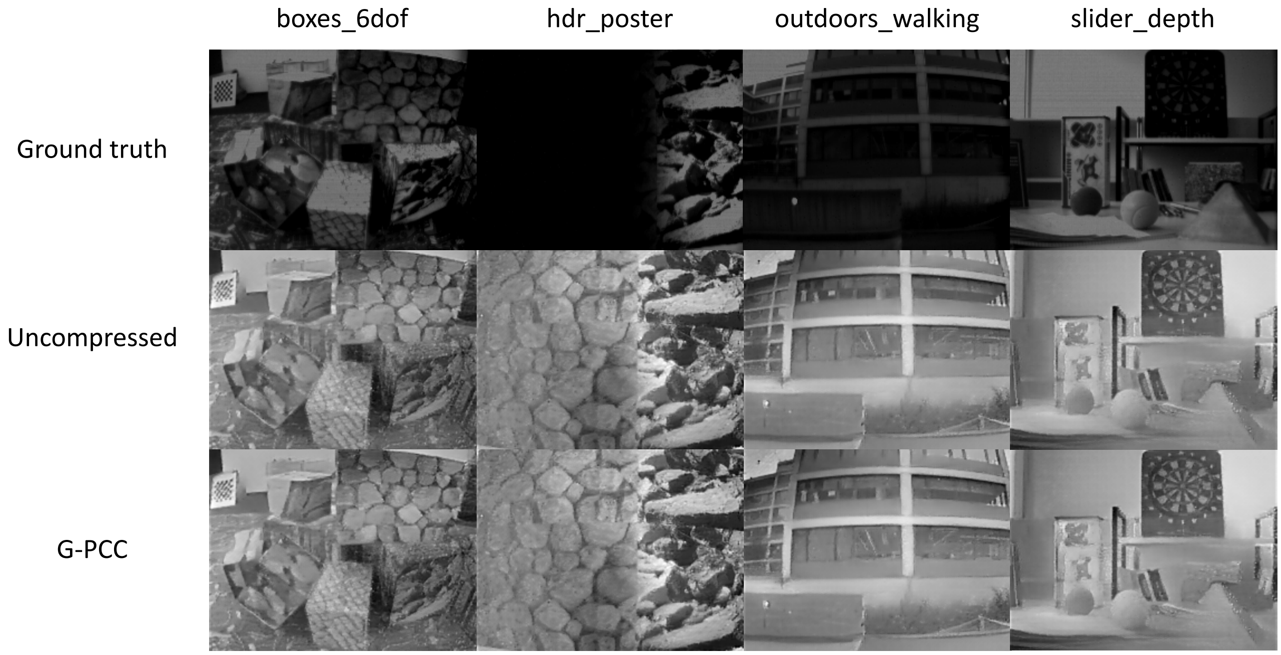

Figure 4. Visualization results of sequences in DAVIS. The G-PCC results are at the highest compression ratio.

7. CONCLUSION

In this paper, a point cloud representation-based lossy event data compression method is proposed and its influence on event-based vision tasks is investigated. Our method restructures the event data stream into point cloud formations, compresses them using point cloud coding in a lossy manner, and the decompressed event point clouds are further tested across four event-based vision tasks, encompassing a range of domains, including visual information interpretation, reconstruction, motion analysis, and 3D vision. Based on our empirical findings, it is ascertained that high compression ratios via lossy compression can be realized while maintaining tolerable performance degradation in event-based vision tasks. Importantly, the selection of the compression intensity dictates the resultant quality of outputs. This research contributes to the foundational understanding of event data compression aimed at machine learning applications and underscores the potential for achieving efficient lossy compression of event data streams.

ACKNOWLEDGMENTS

The authors would like to acknowledge support from the Swiss National Scientific Research project entitled ”Compression of Visual information for Humans and Machines (CoViHM)” under grant number 200020_207918.

REFERENCES

[1] Huang, B. and Ebrahimi, T., “Event data stream compression based on point cloud representation,” in [*2023 IEEE International Conference on Image Processing (ICIP)*], (2023).

[2] Martini, M., Adhuran, J., and Khan, N., “Lossless compression of neuromorphic vision sensor data based on point cloud representation,” *IEEE Access* **10**, 121352–121364 (2022).

[3] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing* **13**(4), 600–612 (2004).

[4] MPEG Systems, "Text of ISO/IEC DIS 23090-18 Carriage of Geometry-based Point Cloud Compression Data." ISO/IEC JTC1/SC29/WG03 Doc. N0075 (Nov. 2020).

[5] Delbruck, T. and Lichtsteiner, P., "Fast sensory motor control based on event-based hybrid neuromorphic-procedural system," in [*2007 IEEE international symposium on circuits and systems*], 845–848, IEEE (2007).

[6] Serrano-Gotarredona, R., Oster, M., Lichtsteiner, P., Linares-Barranco, A., Paz-Vicente, R., Gómez-Rodríguez, F., Camuñas-Mesa, L., Berner, R., Rivas-Pérez, M., Delbruck, T., et al., "Caviar: A 45k neuron, 5m synapse, 12g connects/s aer hardware sensory–processing–learning–actuating system for high-speed visual object recognition and tracking," *IEEE Transactions on Neural networks* **20**(9), 1417–1438 (2009).

[7] Lagorce, X., Orchard, G., Galluppi, F., Shi, B. E., and Benosman, R. B., "Hots: a hierarchy of event-based time-surfaces for pattern recognition," *IEEE transactions on pattern analysis and machine intelligence* **39**(7), 1346–1359 (2016).

[8] Moeys, D. P., Corradi, F., Kerr, E., Vance, P., Das, G., Neil, D., Kerr, D., and Delbrück, T., "Steering a predator robot using a mixed frame/event-driven convolutional neural network," in [*2016 Second international conference on event-based control, communication, and signal processing (EBCCSP)*], 1–8, IEEE (2016).

[9] Pérez-Carrasco, J. A., Zhao, B., Serrano, C., Acha, B., Serrano-Gotarredona, T., Chen, S., and Linares-Barranco, B., "Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward convnets," *IEEE transactions on pattern analysis and machine intelligence* **35**(11), 2706–2719 (2013).

[10] Gehrig, D., Loquercio, A., Derpanis, K. G., and Scaramuzza, D., "End-to-end learning of representations for asynchronous event-based data," in [*Proceedings of the IEEE/CVF International Conference on Computer Vision*], 5633–5643 (2019).

[11] Scheerlinck, C., Barnes, N., and Mahony, R., "Continuous-time intensity estimation using event cameras," in [*Asian Conference on Computer Vision*], 308–324, Springer (2018).

[12] Cook, M., Gugelmann, L., Jug, F., Krautz, C., and Steger, A., "Interacting maps for fast visual interpretation," in [*The 2011 International Joint Conference on Neural Networks*], 770–776, IEEE (2011).

[13] Bardow, P., Davison, A. J., and Leutenegger, S., "Simultaneous optical flow and intensity estimation from an event camera," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 884–892 (2016).

[14] Rebecq, H., Ranftl, R., Koltun, V., and Scaramuzza, D., "Events-to-video: Bringing modern computer vision to event cameras," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 3857–3866 (2019).

[15] Rebecq, H., Ranftl, R., Koltun, V., and Scaramuzza, D., "High speed and high dynamic range video with an event camera," *IEEE transactions on pattern analysis and machine intelligence* **43**(6), 1964–1980 (2019).

[16] Benosman, R., Ieng, S.-H., Clercq, C., Bartolozzi, C., and Srinivasan, M., "Asynchronous frameless event-based optical flow," *Neural Networks* **27**, 32–37 (2012).

[17] Benosman, R., Clercq, C., Lagorce, X., Ieng, S.-H., and Bartolozzi, C., "Event-based visual flow," *IEEE transactions on neural networks and learning systems* **25**(2), 407–417 (2013).

[18] Zhu, A. Z., Yuan, L., Chaney, K., and Daniilidis, K., "Ev-flownet: Self-supervised optical flow estimation for event-based cameras," *arXiv preprint arXiv:1802.06898* (2018).

[19] Zhu, A. Z., Yuan, L., Chaney, K., and Daniilidis, K., "Unsupervised event-based learning of optical flow, depth, and egomotion," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 989–997 (2019).

[20] Ye, C., Mitrokhin, A., Fermüller, C., Yorke, J. A., and Aloimonos, Y., "Unsupervised learning of dense optical flow, depth and egomotion from sparse event data," *arXiv preprint arXiv:1809.08625* (2018).

[21] Wan, Z., Dai, Y., and Mao, Y., "Learning dense and continuous optical flow from an event camera," *IEEE Transactions on Image Processing* **31**, 7237–7251 (2022).

[22] Kogler, J., Sulzbachner, C., Humenberger, M., and Eibensteiner, F., "Address-event based stereo vision with bio-inspired silicon retina imagers," *Advances in theory and applications of stereo vision* , 165–188 (2011).

[23] Ieng, S.-H., Carneiro, J., Osswald, M., and Benosman, R., "Neuromorphic event-based generalized time-based stereovision," *Frontiers in neuroscience* **12**, 442 (2018).

[24] Mahowald, M., "Vlsi analogs of neuronal visual processing: a synthesis of form and function," (1992).

[25] Osswald, M., Ieng, S.-H., Benosman, R., and Indiveri, G., "A spiking neural network model of 3d perception for event-based neuromorphic stereo vision systems," *Scientific reports* **7**(1), 40703 (2017).

[26] Marr, D. and Poggio, T., "Cooperative computation of stereo disparity: A cooperative algorithm is derived for extracting disparity information from stereo image pairs.," *Science* **194**(4262), 283–287 (1976).

[27] Kogler, J., Eibensteiner, F., Humenberger, M., Sulzbachner, C., Gelautz, M., and Scharinger, J., "Enhancement of sparse silicon retina-based stereo matching using belief propagation and two-stage postfiltering," *Journal of Electronic Imaging* **23**(4), 043011–043011 (2014).

[28] Xie, Z., Chen, S., and Orchard, G., "Event-based stereo depth estimation using belief propagation," *Frontiers in neuroscience* **11**, 535 (2017).

[29] Brebion, V., Moreau, J., and Davoine, F., "Learning to estimate two dense depths from lidar and event data," in [*Scandinavian Conference on Image Analysis*], 517–533, Springer (2023).

[30] Bi, Z., Dong, S., Tian, Y., and Huang, T., "Spike coding for dynamic vision sensors," in [*2018 Data Compression Conference*], 117–126 (2018).

[31] Khan, N., Iqbal, K., and Martini, M. G., "Time-aggregation-based lossless video encoding for neuromorphic vision sensor data," *IEEE Internet of Things Journal* **8**(1), 596–609 (2021).

[32] Schiopu, I. and Bilcu, R. C., "Lossless compression of event camera frames," *IEEE Signal Processing Letters* **29**, 1779–1783 (2022).

[33] Banerjee, S., Wang, Z. W., Chopp, H. H., Cossairt, O., and Katsaggelos, A. K., "Lossy event compression based on image-derived quad trees and poisson disk sampling," in [*2021 IEEE International Conference on Image Processing (ICIP)*], 2154–2158 (2021).

[34] Graziosi, D., Nakagami, O., Kuma, S., Zaghetto, A., Suzuki, T., and Tabatabai, A., "An overview of ongoing point cloud compression standardization activities: video-based (v-pcc) and geometry-based (g-pcc)," *APSIPA Transactions on Signal and Information Processing* **9**, e13 (2020).

[35] Lagorce, X., Orchard, G., Galluppi, F., Shi, B. E., and Benosman, R. B., "Hots: A hierarchy of event-based time-surfaces for pattern recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**(7), 1346–1359 (2017).

[36] He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition," in [*2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 770–778 (2016).

[37] Menze, M., Heipke, C., and Geiger, A., "Joint 3d estimation of vehicles and scene flow," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **II-3/W5**, 427–434 (2015).

[38] Zhu, A. Z., Thakur, D., Özaslan, T., Pfrommer, B., Kumar, V., and Daniilidis, K., "The multivehicle stereo event camera dataset: An event camera dataset for 3d perception," *IEEE Robotics and Automation Letters* **3**(3), 2032–2039 (2018).

[39] Mueggler, E., Rebecq, H., Gallego, G., Delbruck, T., and Scaramuzza, D., "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM," *The International Journal of Robotics Research* **36**, 142–149 (feb 2017).