EVENT DATA STREAM COMPRESSION BASED ON POINT CLOUD REPRESENTATION

Bowen Huang and Touradj Ebrahimi

Multimedia Signal Processing Group (MMSPG)
École Polytechnique Fédérale de Lausanne (EPFL)
CH-1012 Lausanne, Switzerland

ABSTRACT

The Dynamic Vision System (DVS) is a novel image acquisition system that works only when there is a brightness change in a pixel, resulting in a stream of events including timestamps, spatial coordinates and the sign of the brightness change (increase or decrease). Although DVS's output data size is much smaller than conventional image systems, it still requires further compression, as the main applications of DVS are embedded systems with limited transmission and storage resources. In this paper, we propose a new method for lossless compression of event data streams based on point cloud representations. The event data stream is organized into a 3D point cloud to which a compression algorithm is applied. In addition, different generation strategies are devised in order to compare the compression performance of the proposed approach. Experimental results show an improved compression ratio of about 22% under lossless conditions.

Index Terms— Event camera, data compression, point cloud

1. INTRODUCTION

Event cameras, also known as Dynamic Vision Systems (DVS), are among some of the most promising new image sensors. Instead of recording entire pixels at a fixed frame rate, DVS records changes in light intensity when they occur. This bio-inspired sensing approach enables DVS to capture, more efficiently, contents with high dynamic range, and rapidly moving objects, while saving energy and bandwidth. Such features open up new possibilities in computer vision[1, 2, 3].

The most fundamental novelty behind DVS resides in the idea that information is represented as a succession of 'events' rather than full two-dimensional images captured at a defined frame rate. The events represent pixels with a change in their light intensity and consist of their coordinates, a timestamp, and a polarity indicating the increase or decrease in intensity. To represent visual information as captured by DVS, formats, such as the Address Event Representation (AER)[4], have been defined. In the AER protocol, events (x, y, t, p) are usually represented in 96 or 64 bits, depending on the desired temporal resolution. Event cameras can record events at a frequency of $10^6/s$ and as the resolution of DVS increases from the early implementation exhibiting 128×128 to more recently 640×480 and soon much larger, the output volume of information from DVS becomes a huge burden on the limited transmission and storage resources of embedded systems. Therefore, it is important to find efficient event data stream compression methods for DVS applications. In this paper, we propose a straightforward and efficient approach to lossless compression by using point cloud representation, exploiting both spatial and temporal redundancies in event data streams.

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.

2. RELATED WORK

2.1. General Purpose Methods

As continuous event data can be considered as a stream of AER packets, generic data compression algorithms such as arithmetic coding, dictionary coding, and fast integer compression can be used to compress event stream data. Arithmetic coding is quite a versatile lossless data compression approach and used to reduce the volume of any type of data based on its statistics. It is widely used as entropy coding methods in most compression systems, such as HEVC and G-PCC. The dictionary encoding strategy maintains a dictionary for the different symbols and the encoder attemps to find a match between the existing items in a dictionary and the input symbols. If successful, a shorter string can be produced in place of the input symbol. Fast integer compression, such as SIMDBP128 developed by Google[5], is used to compress large arrays of integers in search engine and database applications. Fast integer compression is known for its excellent encoding and decoding speed.

Constrained by hardware limitations and high requirements for real-time processing, the authors in[6] proposed a compression algorithm for IoT devices, dubbed Sprintz, suitable for streaming data with high correlation, requiring very low memory (less than 1KB) and offering very low latency.

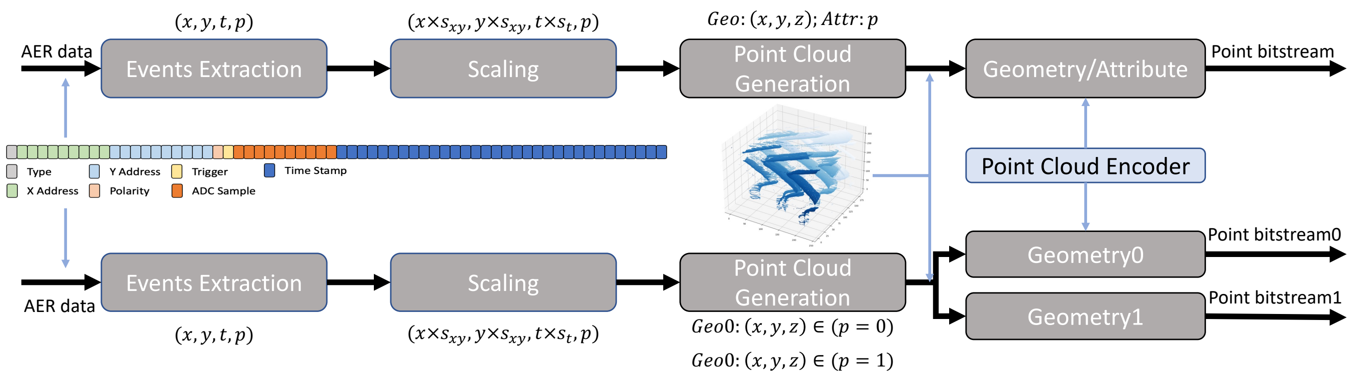


Fig. 1. The framework of the proposed method. The upper pipeline generates one event point cloud and regards the polarity as an extra attribute; the lower pipeline splits input event data into two according to the polarity and positive event point cloud and negative event point cloud are compressed separately.

2.2. DVS Specific Methods

In order to make full use of the characteristics of DVS data streams, DVS-specific compression methods have been developed. A lossless compression method is proposed in [7], where authors devise two different modes, address-first (AP) and time-first (TP), to handle spatially dispersed and spatially concentrated data streams, respectively. An context-adaptive binary arithmetic coding (CABAC) is then cascaded to produce the final bit stream.

Among the specially designed methods, temporal aggregation is a popular strategy where events at a fixed time interval are aggregated into a 2D event frame (EF) and a conventional image codec is applied to compress the EFs[8, 9]. The main disadvantage of EFs is that their aggregation interval is fixed, which does not meet the different requirements of the various temporal resolutions in CV tasks. For example, for motion estimation tasks in autonomous driving, the optimal time interval for spikes is $\approx$ 50 ms. Whereas for object detection, an interval of $\approx$ 10 ms seems to be a better choice, as more detailed features are required[8]. In addition, the sparsity of DVS data makes traditional video codecs inefficient as they are designed for dense pixels.

As event data has been proven to be spatially and temporally correlated[7, 10], we propose to convert event data sequences to event point clouds and use point cloud coding to compress the data. The sparsity of event data sequences is similar to that of point clouds, which lends itself well to point cloud coding. A similar approach is presented in [11]. In this paper, different parameters are proposed when generating the event point clouds to reveal the effect of the parameters. Our proposed strategy is described in detail in the following sections.

3. PROPOSED METHOD

The general idea behind our proposed method is to consider the event timestamp t as a third axis and to compress events during certain time intervals as a point cloud with coordinates (x, y, t) using a point cloud coding approach. The framework is shown in Fig.1. Although the idea is intuitive, several factors need to be investigated to achieve better compression efficiency. In this section, we first present related insights on point cloud coding (PCC) and then describe the key steps of our proposed approach.

3.1. Point Cloud Coding

A point cloud is a data structure containing various separate points with coordinates, called geometry information, and other characteristics such as color or reflectance, called attributes. In order to improve storage and transmission performance and to make more efficient use of the data, point cloud coding methods are proposed to reduce redundancy in 3D space.

The Moving Picture Experts Group (MPEG) has been working on the standardization of PCC since 2018, and two methods have been developed for three different scenarios in point cloud applications[12]. One is Video-based PCC (V-PCC), which is based on 3D to 2D projection and utilisation of 2D video coding . The other is G-PCC, which differs from V-PCC in that it directly encodes the point cloud by using an octree structure. G-PCC mainly focuses on the irregular point clouds with more sparsity, which is similar to the sparsity of event data, and the use of octree increases its applicability to data with similar structures. Some improvements[13, 14] have also been proposed for V-PCC and G-PCC recently.

Another popular PCC method is Draco, a software developed by Google, which can compress point clouds as meshes. Draco can compress arbitrary attributes, making it perfect for special point clouds such as the event point cloud with polar-

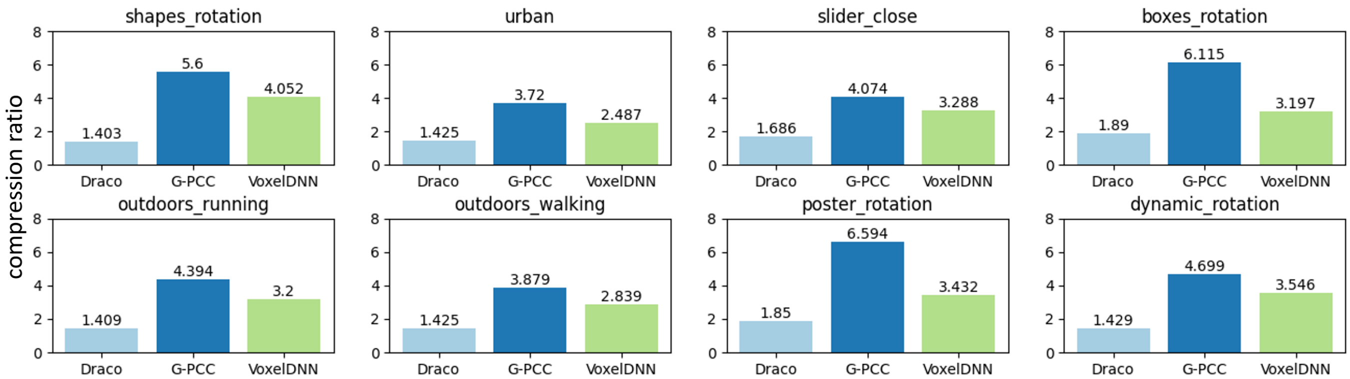


Fig. 2. Comparison of different cascaded PCC.

ity as an additional attribute.

Taking advantage of the development of deep learning methods, neural networks are also introduced in PCC[15, 16, 17, 18]. Auto-encoder structure is commonly used, and the residual block and attention mechanism are added to improve the compression performance. In our experiments, a DNN-based method called VoxelDNN [15] was selected for a more complete comparison. The VoxelDNN encoder operates in an octree and voxel hybrid mode, and geometry information is extracted by convolutional layers. A learning-based context model is used to perform entropy coding.

3.2. Coding Procedure

As shown in Fig.1, the first step of our proposed method is to generate a point cloud representation of the event data in 3D space. The timestamp of the spike can be considered as a third dimension. Since the time resolution of the event data is inconsistent with the spatial coordinates in the order of magnitude, the smallest time stamp is subtracted from others and each time stamp is multiplied by a scaling coefficient. The polarity can be stored as an attribute and compressed with the geometry. The event data can also split into two parts with positive and negative polarities and be compressed separately. The effect of these two-generation strategies will be evaluated by experiments.

The problem of how to control the size of the event point cloud is also considered. Events can be accumulated by a fixed number, resulting in an event point cloud with a fixed number of points, or by a fixed time interval, resulting in an event point cloud of fixed three-dimensional size. Experiments have been carried out to also assess the influence of each alternative.

The coding procedure of the proposed method is summarised as follows. The input event data stream is aggregated either by a fixed number of points or by a fixed time interval to generate a three-dimensional point cloud with coordinates (x, y, t). Scaling is then applied to make the range of coordinates in the generated event point cloud reasonable. Once generated, a point cloud coding is used to compress the event point clouds, and the output is combined with overhead information regarding scaling factor, etc.

4. EXPERIMENTS

4.1. Comparison of Cascaded PCC

Dynamic and Active-pixel Vision Sensor (DAVIS) dataset[19] was used in the experiments. The spatial resolution of the dataset is 180×240, while the temporal resolution is as high as $1 \times 10^6/s$. Four indoor sequences and four outdoor sequences were selected. The complexity of selected sequences, represented by the number of events per second (kev/s), is listed in Table1. Without further explaination, in the following experiments, the temporal scaling factor is set to 1×10^6 to ensure lossless compression, and to keep the consistence of the order of magnitude, (x, y) is multiplied by a spatial scaling factor 1×10^3.

Two conventional and one DNN-based PCC approaches were chosen to perform a comprehensive analysis of the impact of the cascaded PCC. The VoxelDNN is retrained on a training data set generated by DAVIS, with different contents from the test sequences in experiments. The result of the comparison is shown in Fig.2. As presented, G-PCC achieves the highest compression ratio and VoxelDNN also achieves good compression results due to the powerful CNN in selecting suitable filters. However, it is observed that VoxelDNN suffers from slow coding speed, while Draco is the fastest. Since G-PCC achieves a good balance between compression ratio and coding speed, G-PCC was selected as the cascaded PCC in further experiments.

4.2. Comparison of Compression Efficiency

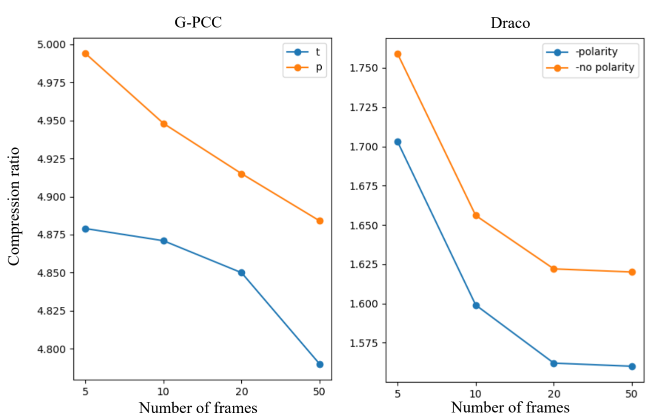
To demonstrate the superiority of the proposed method, our method was compared to SPIKE coding, LZMA and Sprints using the best PCC in 4.1. The results of the benchmark methods are from[20]. Table1 reports the compression efficiency

Table 1. The compression efficiency results (expressed in compression ratio) of different methods.

Sequence Name	Scene Type	Event Rate (kev/s)	Compression Methods					
			Huffman	LZMA	SIMDBP128	Sprintz Delta	Spike Coding	Proposed
shapes	indoor	242.01	1.79	3.04	1.31	2.26	3.78	5.6
boxes	indoor	4288.65	1.96	4.92	1.38	2.83	4.95	6.155
poster	indoor	4021.10	1.96	4.77	1.37	2.76	4.88	6.594
slider	indoor	460.11	1.79	3.19	1.36	2.36	3.84	4.074
dynamic	outdoor	867.24	1.89	3.34	1.28	2.33	3.85	4.699
walking	outdoor	341.64	1.84	3.11	1.31	2.3	3.54	3.879
running	outdoor	713.68	1.87	3.25	1.32	2.33	3.68	4.394
urban	outdoor	478.14	1.83	3.13	1.35	2.31	3.45	3.72
Average		1426.57	1.87	3.59	1.34	2.44	4.00	**4.889**

results. The comparison is made under lossless conditions, which distinguishes our approach from those based on EF with a temporal aggregation step. The best average result is highlighted.

In the comparison, one can distinguish that the proposed method outperforms the others in terms of compression efficiency for all tested sequences. It can also be observed that the proposed method is more suitable for indoor sequences when compared to outdoor sequences. The reason is that outdoor scenes are more complex than indoor scenes, and the event rate cannot fully reveal the complexity of the scene, because fast-moving objects also generate a large number of events regardless of the content of the scene. This phenomenon hints into potential future improvements in outdoor scenarios.



(a) p/t strategy comparison (b) polarity/no polarity strategy comparison

Fig. 3. Comparison of different aggregation strategies.

4.3. Comparison of Aggregation Strategy

As discussed in the previous section, two aggregation strategies are considered, one according to a fixed number of event points, and the other according to a fixed time interval. In the experiments, the input event stream is split into a different number of frames using different strategies prior to lossless compression. The comparison results are shown in Fig.3(a), where ”-t” means fixed time interval and ”-p” means fixed-point.

In Fig.3(a), one can find that the ”-p” strategy achieves a better performance. The size of the event point cloud generated by ”-t” strategy is largely influenced by the content of the scene, while the ”-p” strategy leads to stable number of points and similar spatial and temporal redundancy, beneficial for point cloud compression.

The segmentation of the polarity also has an effect in the performance as experiments in Fig.3(b) show. It can be observed that the compression performance of event point clouds without polarity segmentation is better with Draco, probably because the increase in the number of points mitigates sparsity, leading to a better compression efficiency.

Finally, as shown in the figure, when the number of frames increases, the number of points per frame decreases, and the average compression ratio also decreases. This can be explained by the fact that more temporal redundancy is included with more points, resulting in a higher overall compression efficiency.

5. CONCLUSION

In this paper, a new approach for lossless coding of event data streams is proposed without aggregating events into frames. A point cloud structure is used to represent the event data by relying on point cloud coding. Experiments show that the approach outperforms benchmark values in terms of compression efficiency. The influence of the choice of cascaded point cloud coding, event point cloud generation criteria, and event point cloud size, has been explored and reported, providing a more comprehensive understanding of the proposed approach, and pointing to its potential to efficiently compress event data streams in DVS.

6. REFERENCES

[1] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, 2022.

[2] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck, "A 128$\times$ 128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[3] Guang Chen, Fa Wang, Weijun Li, Lin Hong, Jörg Conradt, Jieneng Chen, Zhenyan Zhang, Yiwen Lu, and Alois Knoll, "Neuroiv: Neuromorphic vision meets intelligent vehicle towards safe driving with a new database and baseline evaluations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1171–1183, 2022.

[4] T. Serrano-Gotarredona, A.G. Andreou, and B. Linares-Barranco, "Aer image filtering architecture for vision-processing systems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 46, no. 9, pp. 1064–1071, 1999.

[5] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 142–149, 2017.

[6] Davis Blalock, Samuel Madden, and John Guttag, "Sprintz: Time series compression for the internet of things," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 3, sep 2018.

[7] Zhichao Bi, Siwei Dong, Yonghong Tian, and Tiejun Huang, "Spike coding for dynamic vision sensors," in *2018 Data Compression Conference*, 2018, pp. 117–126.

[8] N. Khan, K. Iqbal, and M. G. Martini, "Time-aggregation-based lossless video encoding for neuromorphic vision sensor data," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 596–609, 2021.

[9] Ionut Schiopu and Radu Ciprian Bilcu, "Lossless compression of event camera frames," *IEEE Signal Processing Letters*, vol. 29, pp. 1779–1783, 2022.

[10] Siwei Dong, Zhichao Bi, Yonghong Tian, and Tiejun Huang, "Spike coding for dynamic vision sensor in intelligent driving," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 60–71, 2019.

[11] M. Martini, J. Adhuran, and N. Khan, "Lossless compression of neuromorphic vision sensor data based on point cloud representation," *IEEE Access*, vol. 10, pp. 121352–121364, 2022.

[12] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, "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*, vol. 9, pp. e13, 2020.

[13] Ashek Ahmmed, Manoranjan Paul, Manzur Murshed, and David Taubman, "Dynamic point cloud geometry compression using cuboid based commonality modeling framework," in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 2159–2163.

[14] Simone Milani, Enrico Polo, and Simone Limuti, "A transform coding strategy for dynamic point clouds," *IEEE Transactions on Image Processing*, vol. 29, pp. 8213–8225, 2020.

[15] Dat Thanh Nguyen, Maurice Quach, Giuseppe Valenzise, and Pierre Duhamel, "Learning-based lossless compression of 3d point cloud geometry," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 4220–4224.

[16] Jianqiang Wang, Hao Zhu, Haojie Liu, and Zhan Ma, "Lossy point cloud geometry compression via end-to-end learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4909–4923, 2021.

[17] Maurice Quach, Giuseppe Valenzise, and Frederic Dufaux, "Improved deep point cloud geometry compression," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, 2020, pp. 1–6.

[18] André F. R. Guarda, Nuno M. M. Rodrigues, and Fernando Pereira, "Point cloud geometry scalable coding with a single end-to-end deep learning model," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 3354–3358.

[19] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 142–149, feb 2017.

[20] Nabeel Khan, Khurram Iqbal, and Maria G. Martini, "Lossless compression of data from static and mobile dynamic vision sensors-performance and trade-offs," *IEEE Access*, vol. 8, pp. 103149–103163, 2020.