This paper has been accepted for publication at the
IAPR IEEE/Computer Society International Conference on Pattern Recognition (ICPR), Milan, 2021.

Unsupervised Feature Learning for Event Data: Direct vs Inverse Problem Formulation

Dimche Kostadinov, Davide Scaramuzza
Robotics and Perception Group
University of Zurich, Switzerland

***Abstract*—Event-based cameras record an asynchronous stream of per-pixel brightness changes. As such, they have numerous advantages over the standard frame-based cameras, including high temporal resolution, high dynamic range, and no motion blur. Due to the asynchronous nature, efficient learning of compact representation for event data is challenging. While it remains not explored the extent to which the spatial and temporal event "information" is useful for pattern recognition tasks. In this paper, we focus on single-layer architectures. We analyze the performance of two general problem formulations: the direct and the inverse, for unsupervised feature learning from local event data (local volumes of events described in space-time). We identify and show the main advantages of each approach. Theoretically, we analyze guarantees for an optimal solution, possibility for asynchronous, parallel parameter update, and the computational complexity. We present numerical experiments for object recognition. We evaluate the solution under the direct and the inverse problem and give a comparison with the state-of-the-art methods. Our empirical results highlight the advantages of both approaches for representation learning from event data. We show improvements of up to 9 % in the recognition accuracy compared to the state-of-the-art methods from the same class of methods.**

I. Introduction

By asynchronously capturing the light changes in a scene, the event-based camera offers an alternative approach for imaging, which is fundamentally different from the common frame-based cameras. Rather than measuring the “absolute” brightness at a constant rate, the event-based cameras measure per-pixel brightness changes (called “events”) in an asynchronous manner. Some of their main advantages are very high temporal resolution and low latency (both in the order of microseconds), very high dynamic range (140 dB vs. 60 dB of standard cameras), and low power consumption. Hence, event cameras have promising potential for pattern recognition, machine learning, computer vision, robotics, and other wearable applications in challenging scenarios (*e.g.* high-speed motion and the scene has a high dynamic range).

As data-driven sensors, the event-based camera output depends on the brightness change caused by the camera’s motion or the objects’ motion in the scene. The faster the motion, the more events per second are generated since each pixel adapts its sampling rate to the rate of change of the intensity signal that it monitors. One of the critical questions of the paradigm shift posed by event cameras is how to extract meaningful and useful information from the event stream to fulfill a given task. In the past, several unsupervised event-based features were proposed [17] for various tasks, like recognition, object detection, segmentation, and feature tracking. Based on how the feature is estimated, these methods can be grouped on two broad categories: (i) handcrafted and (ii) learned approaches. Concerning the used model, the latter category can be divided into two subgroups: (a) single-layer and (b) multi-layer architecture.

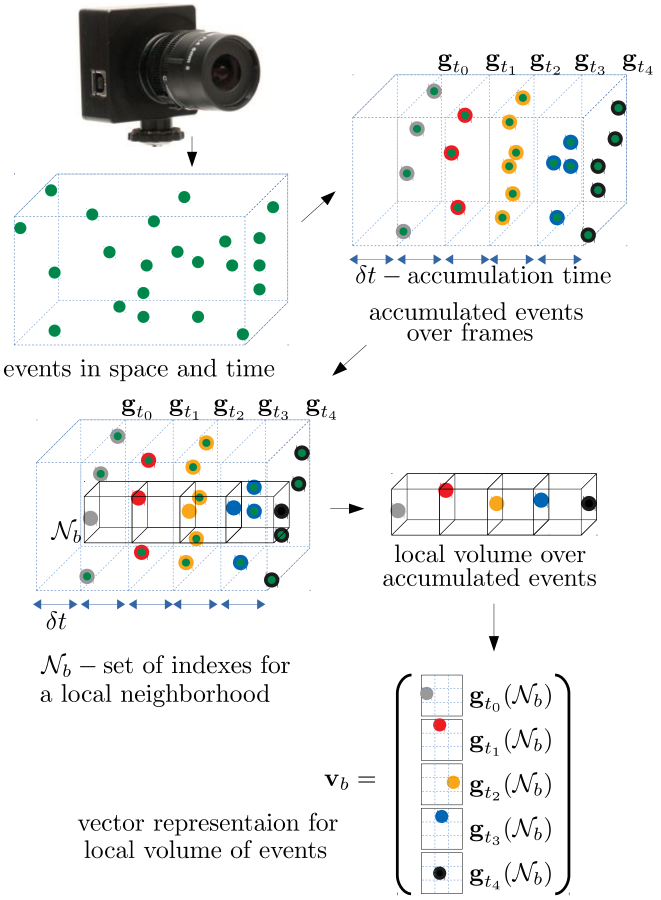
$\mathbf{g}_{t_0}$ $\mathbf{g}_{t_1}$ $\mathbf{g}_{t_2}$ $\mathbf{g}_{t_3}$ $\mathbf{g}_{t_4}$
δt – accumulation time
accumulated events over frames
events in space and time
$\mathcal{N}_b$
δt
local volume over accumulated events
$\mathcal{N}_b$ – set of indexes for a local neighborhood
$\mathbf{v}_b =$
vector representaion for local volume of events
$\mathbf{g}_{t_0}(\mathcal{N}_b)$
$\mathbf{g}_{t_1}(\mathcal{N}_b)$
$\mathbf{g}_{t_2}(\mathcal{N}_b)$
$\mathbf{g}_{t_3}(\mathcal{N}_b)$
$\mathbf{g}_{t_4}(\mathcal{N}_b)$

Fig. 1. An illustration for the construction of the local volume of aggregated events that we use as a vector input representation $\mathbf{v}_b$ in our unsupervised feature learning approach.

Deep, multi-layered architectures have shown to be successful at many tasks, but most of these methods process events in synchronous batches, sacrificing the asynchronous property of event data. Asynchronous and parallel learning for event-based data was addressed under the spiking neural

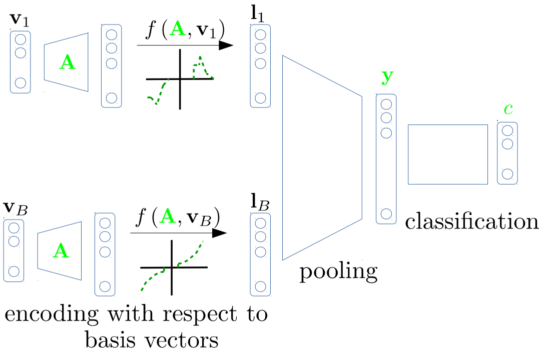
$\mathbf{v}_1$
A
$f(\mathbf{A}, \mathbf{v}_1)$
$\mathbf{l}_1$
y
c
$\mathbf{v}_B$
A
$f(\mathbf{A}, \mathbf{v}_B)$
$\mathbf{l}_B$
classification
pooling
encoding with respect to basis vectors

Fig. 2. A schematic diagram which illustrates the main components in the proposed recognition pipeline.

networks (SNN) [18]. However, SNNs were challenging to train due to the absence of an efficient equivalent to back-propagation learning method and the hyper-parameters' sensitivity. While the interpretations and the understanding of the learning dynamics, even for the most popular multi-layer architectures, remain challenging. On the other hand, data-adaptive, *i.e.*, learned single-layer architectures for event-based data were not studied extensively. Moreover, the analysis for the appropriate problem formulation, which would be advantageous for efficient, asynchronous, and parallel learning from event-based data, was not fully explored. As such, it remains unknown to which extend a single-layer model could be useful for event-based data and how the spatial and temporal resolution of the event-based data impacts performance for a given task.

This paper analyzes two classes of single-layer methods for compact, information preserving, and task-relevant representation learning from event-based data. We focus on unsupervised learning of a set of basis vectors (or filter bank). We encode the event-based input data with respect to the basis vectors to produce features, which we use for recognition. In general, the problem of learning such basis vectors can be formulated as a direct [32], [33], [23] or inverse problem [35], [19] [29].

- Under the inverse problem, we would like to estimate (represent) the input event-data as a linear combination over a given set of basis vectors.
- While under the direct problem, we would like to estimate (represent) the input event-data as a set of projections over a set of basis vectors.

We highlight and show the advantages of both approaches. Theoretically, we reflect on the optimal solution and the complexity for asynchronous and parallel updates under both problem formulations. In both cases, we have to jointly estimate the representation and learn the set of basis vectors during training time. However, under the direct problem formulation, the complexity for estimating the representation is low. While under the inverse problem formulation, the complexity can be high, especially when the input event-based data dimension is high, or the number of basis vectors is high. We evaluate both methods on different data sets for the task of object recognition. Our validation shows improvements in the recognition accuracy while using event-data at low spatial resolution compared to the state-of-the-art methods.

A. Contributions

In the following, we give our main contributions.

- We analyze the recognition performance of a two part recognition pipeline: (i) unsupervised feature learning and (ii) supervised classifier over the encoded features. We show that with this simple single layer architecture we can achieve state-of-the-art result which outperforms handcrafted methods.
- In the unsupervised feature learning, we address the direct and the inverse problem formulation, *i.e.*, data adaptive basis vectors learning with respect to which we encode the event-based data. We investigate the solutions of the both problems, comment on the local optimal guarantees as well as highlight the complexity under asynchronous and parallel solution update.
- We validate both approaches trough a numerical evaluation on different data sets for the task of object recognition. In addition, we also provide an analysis for different trade-offs and highlight the advantages of each approach. We demonstrate that the direct problem formulation has identical recognition performance to the inverse problem formulation, but under the direct problem formulation, the learning complexity is lower. In addition to having local convergence guarantees, the direct problem formulation also has lower complexity for asynchronous and parallel update. Our numerical results, compared to the state-of-the-art methods from the same category show improvements of up to 9% in the recognition accuracy.

II. Related Work

Unsupervised feature learning is a well studied topic, *e.g.*, [28], [13], [39], [24], [38], [43] in the area of image processing, computer vision and machine learning for object recognition, detection and matching. On the other hand, only recently, the event-based vision has been used to address these problems.

Analogously to the approaches used for a standard camera image, feature extraction approaches from event-based camera can be grouped into two broad categories handcrafted and learning-based. Spatio-temporal feature descriptors of the event stream were used for high-level applications like gesture recognition [22], object recognition [27], [4] or face detection [7]. Low-level applications include optical flow prediction [37], [34] and image reconstruction [26]. As a data-driven model, asynchronous, spiking neural networks (SNNs) [21] have been applied to several tasks, *e.g.*, object recognition [21], [15], [20], gesture classification [6], and optical flow prediction [37]. However, computationally efficient equivalent to back-propagation algorithms still limits the usability of SNNs in complex real-world scenarios. Several works have recently proposed using standard learning architectures as an alternative to SNNs [5], [11], [4], [3]. Commonly, these works use a handcrafted event stream representation. Deep multi-layer architectures have shown to be successful at many tasks, but asynchronous and parallel learning and the interpretation and understanding of the learning dynamics remain challenging. While not much attention was given to single-layer architectures that exploit the appropriate problem formulation, which would be advantageous for efficient, asynchronous, and parallel learning from event-based data.

III. Paper Organization

The rest of the paper is organized as follows. In Section IV, we introduce the working principle of the event-based camera. In Section V, we first present an overview of our approach, then we present how we form the input for the unsupervised learning algorithm. Afterward, we give the problem formulation for our unsupervised learning approach and describe our classifier. We devote Section VI to numerical evaluation, while with Section VII, we conclude the paper.

IV. Event Based Camera Working Principle

In this section, we present the working principle of event cameras. The event-based cameras (like DVS [1]) at independent pixels locations measure "events" according to brightness change. Let $L([\begin{smallmatrix} x \\ y \end{smallmatrix}], t) = \log I([\begin{smallmatrix} x \\ y \end{smallmatrix}], t)$ be the logarithmic brightness at pixel location $[\begin{smallmatrix} x \\ y \end{smallmatrix}]$ on the image plane. The event-based camera generates an event $\mathbf{e}_t = \{[\begin{smallmatrix} x \\ y \end{smallmatrix}], t_k, p_k\}$ when the change in logarithmic brightness at pixel location $[\begin{smallmatrix} x \\ y \end{smallmatrix}]$ reaches a threshold C, *i.e.*,

$$\Delta L = L([\begin{smallmatrix} x \\ y \end{smallmatrix}], t_k) - L([\begin{smallmatrix} x \\ y \end{smallmatrix}], t - \Delta t) = p_t(x, y) C, \tag{1}$$

where t is the time "stamp" of the event, Δt is the time since the previous event at the same pixel location $[\begin{smallmatrix} x \\ y \end{smallmatrix}]$ and $p_t(x, y) \in \{+1, -1\}$ is the event polarity (i.e., sign of the brightness change). It is important to highlight that an event camera does not produce images at a constant rate, but rather a stream of asynchronous, sparse events in space and time. That is, depending on the visual input, the event-based camera, outputs data proportionally to the amount of brightness changes in the scene.

V. Unsupervised Representation Learning for Event Based Data

This section describes our approach, which focuses on learning features from local volumes of events. We adopt a multi-stage approach, which is similar to those employed in computer vision, as well as other feature learning works [9].

A. Approach Overview

Our approach consists of two parts (i) feature representation learning and (ii) classification. In Figure 2, we show the schematic diagram, which illustrates the main components in the proposed recognition pipeline. The unsupervised learning part includes the following steps:

1) Extract random local volumes of events from unlabeled set of events for training and apply a pre-processing to the local volume of events.
2) Learn a feature-mapping using an unsupervised learning algorithm.

We address the problem of learning the feature mapping under unsupervised learning algorithm. We highlight that depending on the input representation, our single layer unsupervised feature learning approach has flexibility to accommodate and capture different features. That is, when the local spatial dimension equals to the spatial dimension of the event-based camera stream then our single layer architecture equals to a fully connected layer. In that case, our unsupervised learning of basis vectors consists of learning the weights in the equivalent fully connected layer. On the other hand, when the local spatial dimension is smaller to the spatial dimension of the event-based camera stream and the basis vectors are shared for all local volumes of accumulated events then our single layer architecture equals to a convolutional layer. Furthermore, if the length of the local volume of accumulated events equals to the total number of accumulation intervals then we have 2D convolution. This is indeed the case, since, we actually preform a convolution operation only over the spatial domain, *i.e.*, the X and Y axis. Otherwise, when the length of the local volume of accumulated events is smaller to the total number of accumulation intervals, then we have additional temporal dimension over which we can preform the convolution operation and thus have a 3D convolution. In this paper, we focus on learning a basis vectors that correspond to learning a 2D convolutional filters.

In the second part, given the learned feature mapping and a set of labels for the training events, we perform feature extraction and classification as follows:

1) Extract features from equally spaced local volumes of events covering the input event set and pool features together over regions of the input events to reduce the dimensionality of the feature vector.
2) Train a linear classifier to predict the labels given the feature vectors.

B. Formation of Local Volumes of Events and Pre-processing

Given an set of events across time as input, we extract and use local volumes of accumulated events. This representation is a vector $\mathbf{v}_b \in \Re^{B_x B_y T_l}$ that describes a local volume indexed by b, which has height B_x, width B_y and length T_l that equals to the number of time intervals over which we accumulate events. We construct $\mathbf{v}_b$ as follows:

$$\mathbf{v}_b = \begin{bmatrix} \mathbf{g}_{t_l}(\mathcal{N}_b) \\ \mathbf{g}_{t_{l+1}}(\mathcal{N}_b) \\ \vdots \\ \mathbf{g}_{t_{l+T_l}}(\mathcal{N}_b) \end{bmatrix}, \text{ where } \mathbf{g}_t = \sum_{t}^{t+\delta t} \begin{bmatrix} p_t(1,1) \\ p_t(1,2) \\ \vdots \\ p_t(N_x, N_y) \end{bmatrix}, \tag{2}$$

while $t \in \{0, ..., T\}$ is the time index, δt is the duration of one accumulation interval, $\mathcal{N}_b$, $b \in \{1, ..., B\}$ is the index for a spatial block with size $B_x \times B_y$, centered at spatial spatial position (x, y). An illustration for the construction of the local volume of accumulated events is shown in Figure 1.

Our pre-processing consists of two parts. In the first part, we normalize each local volume of accumulated events $\mathbf{v}_b$ by subtracting the mean and dividing by the standard deviation of its elements. In fact, this corresponds to normalization in the change of the event accumulation. After normalizing each input vector, in the second part, we whiten [9] all $\mathbf{v}_b$ from the entire data set of events. This process is commonly used in the learning methods (e.g., [9]) for standard images, but it is less frequently employed in pattern recognition from event-based data.

C. Unsupervised Feature Learning

Our training data, *i.e.*, $\mathbf{V} = [\mathbf{V}_1, ..., \mathbf{V}_M]$ consists of set of vectors $\mathbf{V}_j = [\mathbf{v}_{j,1}, ..., \mathbf{v}_{j,B}]$, where $\mathbf{v}_{j,b} \in \Re^{B_x B_x T_l}$ represents the local volume of temporally aggregated events over local spatial region indexed by b for the event data j. We consider the direct and inverse problem formulations, where we jointly estimate the representations and learn a set of basis vectors. In the following, we present the both problem formulations.

1) Inverse Problem Formulation: The inverse problem formulation has the following from:

$$\Big[\mathbf{L}, \mathbf{D}\Big] = \arg\min_{\mathbf{L},\mathbf{D}} \frac{1}{2}\|\mathbf{V} - \mathbf{D}\mathbf{L}\|_F^2 + \lambda_0 m(\mathbf{L}) + \lambda_1 \Omega(\mathbf{D}), \quad (3)$$

where $\|.\|_F$ denotes Frobenius norm, $m(\mathbf{L}) = \sum_{j=1}^{M}\sum_{b=1}^{B}\|\mathbf{l}_{j,b}\|_1$ and $\Omega(\mathbf{D}) = \lambda_2\|\mathbf{D}\|_2^2 - \lambda_3\|\mathbf{D}\mathbf{D}^T - \mathbf{I}\|_2^2 - \lambda_4 \log|\det \mathbf{D}\mathbf{D}^T|$ are constraints on the representations $\mathbf{L} = [\mathbf{L}_1, ..., \mathbf{L}_M]$, $\mathbf{L}_j = [\mathbf{l}_{j,1}, ..., \mathbf{l}_{j,B}]$, $\mathbf{l}_{j,b} \in \Re^K$ and the dictionary (the set of basis vectors) $\mathbf{D} = [\mathbf{d}_1, ..., \mathbf{d}_K]$, $\mathbf{d}_k \in \Re^{B_x B_x T_l}$ while λ_0 and λ_1 are Lagrangian parameters. Given the dictionary $\mathbf{D}$ under the inverse problem (3), we would like to represent the local volume of events $\mathbf{v}_{j,b}$ as a sparse linear combination $\mathbf{l}_{j,b}$, *i.e.*, $\mathbf{v}_{j,b} = \mathbf{D}\mathbf{l}_{j,b}$.

2) Direct Problem Formulation: The direct problem formulation has the following from:

$$\Big[\mathbf{L}, \mathbf{A}\Big] = \arg\min_{\mathbf{L},\mathbf{A}} \frac{1}{2}\|\mathbf{A}\mathbf{V} - \mathbf{L}\|_F^2 + \lambda_0 m(\mathbf{L}) + \lambda_1 \Omega(\mathbf{A}), \quad (4)$$

where $\mathbf{A} = \begin{bmatrix} \mathbf{a}_1^T \\ \cdot \\ \cdot \\ \mathbf{a}_K^T \end{bmatrix}$, $\mathbf{a}_k \in \Re^{B_x B_x T_l}$, while the constraints $m(\mathbf{L})$ and $(\mathbf{A})$ are equivalent with the ones defined for (3). Given the linear map $\mathbf{A}$, under the direct problem (4), we would like to represent $\mathbf{v}_{j,b}$ as a two step nonlinear transform $\mathbf{l}_{l,b} = g(\mathbf{A}\mathbf{v}_{l,b})$ consisting of: (i) linear mapping $\mathbf{A}\mathbf{v}_{j,b}$ and (ii) element-wise nonlinearity $g(\mathbf{A}\mathbf{v}_{l,b})$, which is induced by the constraint $m(\mathbf{l}_{j,b}) = \|\mathbf{l}_{j,b}\|_1$.

Both, (3) and (4) are non-convex in the variables $\{\mathbf{L}, \mathbf{D}\}$ and $\{\mathbf{L}, \mathbf{A}\}$, respectively. If the variable $\mathbf{D}$ in (3) (or $\mathbf{A}$ in (4)) is fixed, (3) (or (4)) is convex, but if $\mathbf{L}$ is fixed, the reduced problem for (3) (or (4)) might not be convex due to the penalty function Ω. Nonetheless, to solve (3) (or (4)) usually an iterative, alternating algorithm is used that has two steps: dictionary $\mathbf{D}$ (or transform $\mathbf{A}$) update and sparse coding.

Method	Local Convergence Guarantee Under Sparsity Constraints
Proposed (inverse)	exists
Proposed (direct)	exists

TABLE I
THE LOCAL CONVERGENCE GUARANTEE.

Considering the inverse problem (3), in the dictionary update step, given $\mathbf{L}$ that is estimated at iteration t, we use a K-SVD [36]. In the sparse coding step, given $\mathbf{D}^{t+1}$, the sparse codes $\mathbf{l}_{l,b}^{t+1}$ are estimated using [40].

Considering the direct problem (4), in the transform estimate step, given $\mathbf{L}$ that is estimated at iteration t, we use approximate closed form solution to estimate the transform matrix $\mathbf{A}^{t+1}$ at iteration $t+1$. In the sparse coding step, given $\mathbf{A}^{t+1}$, the sparse codes $\mathbf{l}_{j,b}^{t+1}$ are estimated by a closed form solution.

3) Local Convergence Guarantees: Under sparsity constraints for the representations, and conditioning and coherence constraints [23] for the dictionary, a local convergence guarantee to both (3) and (4) has been shown [35], [29] and [23]. However, it is important to note that under the direct problem formulation the complexity of the sparse coding step has very low computational complexity which is linear in the dimension of the representation $\mathbf{l}_{j,b}$, *i.e.*, $O(B_x B_y T_l)$. While for the inverse problem the same complexity is higher [31], [32]. In addition, as advantage, the direct problem allows posing a class of penalties under which a low complexity closed form solution exists.

Method	Complexity Under Sparsity Constraint
Proposed (inverse) Proposed (direct)	$O(sB_x B_y T_l K)$ $O(B_x B_y T_l)$
Method	Complexity Under Broad Class of Constraint
Proposed (inverse) Proposed (direct)	high low

TABLE II
THE COMPLEXITY FOR ESTIMATING THE UNSUPERVISED REPRESENTATION UNDER SPARSITY CONSTRAINTS AND UNDER BROAD CLASS OF CONSTRAINTS.

D. Final Feature Composition and Classification

Under either the direct or inverse problem formulation, we estimate basis vectors, which we consider as the parameters of a function that maps the input local volume of events to a new representation. We apply this mapping to our (labeled) training event-based data for classification.

1) Final Feature Composition: We consider the learned basis vectors as the parameters of an encoding function $f : R^{B_x B_y T_l} \rightarrow R^K$, which represents our feature extractor. Many functions might be used to encode with respect to the learned basis vectors, here we use the triangle encoding, as presented by [9]. Under our encoding function f, for any B_x-by-B_y-by-T_l local volume of accumulated events $\mathbf{v}_{j,b}$, we compute the corresponding representation $\mathbf{l}_{j,b} \in R^K$. Moreover, we define a (single layer) representation for the set of events by applying the function f to all of the local volumes of accumulated events. That is, given the set of events defined over N_x-by-N_y-by-T volume of event locations, for each of the local volumes $\mathbf{v}_{j,b}$ described by the spatial index $b \in \mathcal{N}_b$, we compute the representation $\mathbf{l}_{j,b}$. More formally, we let $\mathbf{l}_{j,b}$ to be the K-dimensional representation extracted for location index b, from the input set of events indexed by j.

We reduce the dimensionality of the event-based data representation by pooling, which is similar as proposed by [9], but instead of image patches, we operate on local volumes of accumulated events and construct the final representation $\mathbf{y}_j = \begin{bmatrix} \sum_{b_1} \mathbf{l}_{j,b_1} \\ \vdots \\ \sum_{b_4} \mathbf{l}_{j,b_4} \end{bmatrix} \in \Re^{4K}$.

2) Classification: In our classification, we use the pooled feature vectors $\mathbf{y}_j$ for each training event-based data and its corresponding label. We apply (L2) SVM [42] classification, with regularization parameter that is determined by cross-validation.

E. Asynchronous and Parallel Update in Feature Learning

We note that the differences between the inverse problem (3) and the direct problem (4) emerge only if the set of basis vectors is over-complete or under-complete. Since, under an orthonormal set of basis vectors the two problems are equivalent. Considering an over-complete set of basis vectors, the solution in the sub-problems related to both the inverse problem (3) and the direct problem (4) have major impact on the possibility for asynchronous and parallel update for the proposed unsupervised feature learning approach.

During the sparse coding step for the inverse problem formulation (3), at any change (even small) of the input representation, a solution to inverse problem has to be estimated, which would lead to high computational complexity and challenges in parallel update of the representation. In contrast, the same step in the direct problem formulation (4) has a closed from solution. Moreover, under (4), it is straightforward to update each element of the representation in parallel and in an asynchronous fashion, with no additional increase in the computational complexity.

During the basis set update, under the conditioning and coherence constraints, for both the inverse (3) and the direct (4) problem formulation, parallel update is challenging. Additional, structure enforcing constraint on the basis set might be helpfully towards parallel and asynchronous update of the basis set, especially under the direct (4) problem formulation.

VI. Numerical Evaluation

In this section, we evaluate the potential of our approach and provide comparative results between our algorithm and the state-of-the-art methods. We consider the task of object recognition over three publicly available data sets. In the following subsection, we describe the setup for the preformed experiments and finally present and discuss the results.

A. Data Sets, Setup and Analysis

In our evaluation, we used N-MNIST [25], N-Caltech101 [25] and N-Cars [1] event-based data sets. We use our classification pipeline, which we presented in the earlier section. We learned the over-complete basis set under the direct and the inverse problem formulation. As an encoding function, we use the triangle encoding as proposed in [8], [9], by which we encode the local volume of events with respect to the learned set of basis vectors. After the puling stage over all the encoded representations for the local volumes of events, we use the final representation to learn a linear SVM classifier.

Considering the N-MNIST data set, we use the final representations from the training data set in order to learning the classifier, while we use the final representations from the test data set for evaluation of the recognition accuracy. Considering the N-Caltheh101 and N-Cars data sets, we use 80% of the data set for training and the remaining to test the recognition performance.

We compare the results of our approach under the direct and the inverse problem formulation for learning the basis set with the state-of-the-art methods. Our evaluation includes comparison to the methods under single layer as well as multi-layer architectures which use event-based data. In addition, we analyze the impact on the recognition performance when we change different parameters in the components of our pipeline. We present the recognition results under varying (i) number of basis vectors, (ii) size of the local volume and (iii) number of accumulation bins.

Method	Acc. %
Hfist [14]	06.0
HOTS [2]	21.0
Garbor-SNN [41]	19.2
HATS [4]	64.2
DART [30]	70.3

Method	Acc. %
EST [16]	81.7
VID2E [10]	90.1

Method	Acc. %
Proposed (inverse)	**78.4**
Proposed (direct)	**77.1**

TABLE III
A comparison of the average precision accuracy for our approach and the state-of-the-art single layer and multi-layer methods, which use event data as input.

Data set	Acc. % (inverse)	Acc. % (direct)
N-MNIST	98.1	96.8
N-Calteh101	78.4	77.1
N-CARS	84.7	81.3

TABLE IV
The average precision accuracy of the proposed method under direct and inverse problem formulation for unsupervised feature learning over the used N-MNIST, N-Calteh101 and N-CARS event-base data sets.

B. Result Discussion

The results of our numerical evaluation are shown in Tables III, IV, and V.

In Tables III and IV, we show a comparison between the average recognition accuracy of our approach under direct and inverse problem formulation and the state-of-the-art single layer methods which use as input event-based data and frame based data. As we can observe, highest accuracy achieves the proposed unsupervised learning approach under inverse problem formulation. The unsupervised learning approach under the direct problem formulation also has high accuracy, which is a bit lower, but the learning complexity is also lower compared to the unsupervised learning approach under inverse problem formulation. Moreover, we can see that the proposed

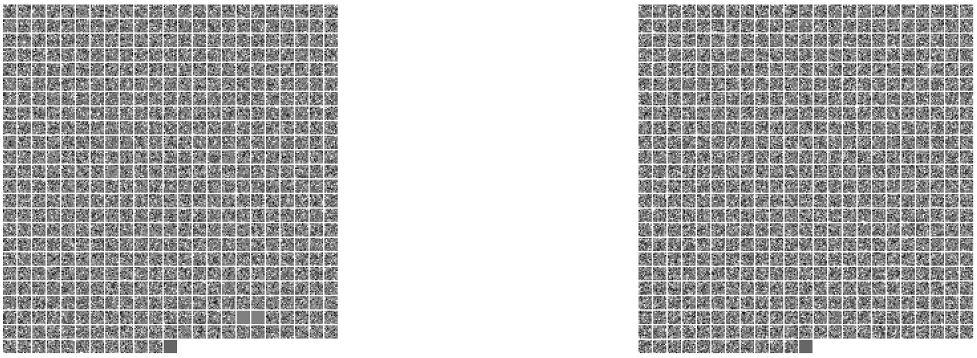

Fig. 3. Visualization for a subset of the learned basis vectors under the inverse and direct problem formulation for the N-CARS data set.

Method	Number of Basis Vectors			
	1000	1500	1700	2000
Proposed (inverse)	73.2	74.5	78.4	76.0
Proposed (direct)	74.3	77.0	77.1	75.5

Method	Size of the Local Volume			
	4×4×4	4×12×12	4×16×16	4×21× 21
Proposed (inverse)	69.6	78.4	76.4	75.2
Proposed (direct)	64.8	77.1	74.5	75.1

Method	Number of Accumulation Intervals			
	2	4	7	10
Proposed (inverse)	61.7	72.4	78.4	76.3
Proposed (direct)	63.2	69.1	77.1	74.1

TABLE V
THE AVERAGE PRECISION ACCURACY UNDER VARYING: (I) NUMBER OF BASIS VECTORS, (II) SIZE OF THE LOCAL VOLUME AND (III) NUMBER OF ACCUMULATION INTERVALS.

unsupervised learning approach outperforms the comparing state-of-the-art methods from the same category. It is important to note that under the proposed unsupervised learning approach, we use as input an event data that has $\times 4$ lower spatial resolution. That is, we use $\sim \mathbf{1}$ order of magnitude less event data compared to the state-of-the-art methods.

In Table III, we also show a comparison between the average recognition accuracy of our approach under direct and inverse problem formulation and the state-of-the-art multi-layer methods which use as input event-based data. We can see that compared to the EST [16] method, the accuracy is lower, but only about 3.3%. On the other had, compared to the VID2E [10] multi-layer method, we have bigger gap in the achieved accuracy. However, we have to note that under the multi-layered methods the authors used (i) pre-trained networks (using image-net [12] data set), (ii) additionally simulated (artificially generated) and augmented event-data, while in our approach, we did not used any additionally simulated or external data. The recognition accuracy over each of the event-based data set is shown in Table IV and we can see that the recognition accuracy is high.

In Table V, we show the recognition results under varying different parameters in the components of our recognition pipeline. As we can see the higher number of basis vectors we have higher recognition accuracy, but up to a certain point. We explain this by the fact that to small number of overcomplete basis vectors does not capture well the inherent invariances over all of the local volumes of events, while to big number of overcomplete set of basis vectors reduces the discrimination property. Interestingly, we have similar behavior when we vary the size of the local volume. However, we explain this by the fact that under very small local volumes the space of variation is to small, hence lower distinguishability between the local blocks, which results in low recognition accuracy. On the other hand, under very big local volumes the space of variation is big, which would imply that we have to use a bigger number of basis vectors[1].

Considering the impact of the number of accumulation intervals, we note that we achieve the highest accuracy when we have 7 accumulation intervals. We suspect that the reason for low accuracy at too small or too big numbers of accumulation intervals is due to the fact that our events are triggered by motion, which we do not know in advance. This adds additional variability in the distribution of the events across the spatial domain that can be interpreted as noise corruption.

VII. Conclusion

In this paper, we presented an analysis of the performance of two general problem formulations, *i.e.*, the direct and the inverse, for unsupervised feature learning from local event data. We identified and show the main advantages of each approach theoretically and by a numerical validation.

Empirically, we demonstrated that the direct problem formulation has similar recognition performance to the inverse problem formulation, but the learning complexity is lower under the direct problem formulation. The direct solution to the problem formulation has lower complexity, advantageous for an asynchronous and parallel update. In our numerical evaluation, the comparison with the state-of-the-art methods from the same category showed improvements of up to $\mathbf{9}\%$ in the recognition accuracy while using fewer event data.

We leave the analysis of different encoding functions, the joint learning of features and classifiers, and the analysis for a multi-layered unsupervised feature learning as feature work.

[1] In Fig. 3, we give visualization of a subset of the learned basis vectors for the N-Cars event-based data set under the inverse and direct problem formulation. We note that there is not much of visual difference between the basis vectors when we have regularization for the conditioning and the coherence of the basis set.

REFERENCES

[1] https://www.prophesee.ai/2018/03/13/dataset-n-cars/.

[2] Hots: a hierarchy of event-based time-surfaces for pattern recognition. *TPAMI*, 2017.

[3] Kenneth Chaney Alex Zihao Zhu, Liangzhe Yuan and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *In Robotics: Science and Systems (RSS)*, 2018.

[4] Nicolas Bourdis Xavier Lagorce Amos Sironi, Manuele Brambilla and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. *In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, page 1731–1740, 2018.

[5] Guillermo Gallego Narciso Garcıa Ana I. Maqueda, Antonio Loquercio and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. *In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, page 5419–5427, 2018.

[6] David Berg Timothy Melano Jeffrey McKinstry Carmelo Di Nolfo-Tapan Nayak Alexander Andreopoulos-Guillaume Garreau Marcela Mendoza Jeff Kusnitz Michael Debole Steve Esser Tobi Delbruck Myron Flickner Arnon Amir, Brian Taba and Dharmendra Modha. A low power, fully event-based gesture recognition system. *In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, page 7388–7397, 2017.

[7] Souptik Barua, Yoshitaka Miyatani, and Ashok Veeraraghavan. Direct face detection and video reconstruction from event cameras. *In IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, pages 1–9, 2016.

[8] A. Coates and A. Y Ng. The importance of encoding versus training with sparse coding and vector quantization. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 921–928, 2011.

[9] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. 15:215–223, 2011.

[10] Javier Hidalgo-Carrió Davide Scaramuzza Daniel Gehrig, Mathias Gehrig. Video to events: Recycling video datasets for event cameras.

[11] Michael Pfeiffer Daniel Neil and Shih-Chii Liu. Phased lstm: Accelerating recurrent network training for long or event-based sequences. *In Conf. Neural Inf. Process. Syst. (NIPS)*, page 3882–3890, 2016.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[13] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. Patern Anal. Mach. Intell.*, pages 743–761, 2012.

[14] R. Etienne-Cummings C. Posch N. Thakor G. Orchard, C. Meyer and R. Benosman. Hfirst: A temporal approach to object recognition. 2015.

[15] Ralph Etienne-Cummings Christoph Posch Nitish Thakor Garrick Orchard, Cedric Meyer and Ryad Benosman. Hfirst: A temporal approach to object recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, page 2028–2040, 2015.

[16] Daniel Gehrig, Antonio Loquercio, Konstantinos G. Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. *CoRR*, 2019.

[17] Garrick Orchard-Chiara Bartolozzi Brian Taba Andrea Censi Stefan Leutenegger Andrew Davison Jorg Conradt Kostas Daniilidis Guillermo Gallego, Tobi Delbruck and Da- vide Scaramuzza. Event-based vision: A survey. *arXiv eprints*, abs/1904.08405, 2019. 2.

[18] Dongsung Huh and Terrence J Sejnowski. Gradient descent for spiking neural networks. *In Conf. Neural Inf. Process. Syst. (NIPS)*, page 1440–1450, 2018.

[19] Z. Jiang, Z. Lin, and L. S. Davis. Label consistent k-svd: Learning a discriminative dictionary for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2651–2664, Nov 2013.

[20] Carmen Serrano Bego Na Acha Teresa Serrano-Gotarredona Shouchun Chen Jose A. Perez-Carrasco, Bo Zhao and Bernabe Linares-Barranco. Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward convnets. *IEEE Trans. Pattern Anal. Mach. Intell.*, page 2706–2719, Nov. 2013.

[21] Tobi Delbruck Jun Haeng Lee and Michael Pfeiffer. Training deep spiking neural networks using backpropagation. *Front. Neurosci.*, page 10:508, 2016.

[22] Paul K. J. Park Michael Pfeiffer Chang-Woo Shin-Hyunsurk Ryu Junhaeng Lee, Tobi Delbruck and Byung Chang Kang. Live demonstration: Gesture-based remote control using stereo pair of dynamic vision sensors. *In IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2012.

[23] Dimche Kostadinov, Slava Voloshynovskiy, and Sohrab Ferdowsi. Learning overcomplete and sparsifying transform with approximate and exact closed form solutions. In *7-th European Workshop on Visual Information Processing (EUVIP)*, Tampere, Finland, November 2018.

[24] David G. Lowe. Distinctive image features from scaleinvariant keypoints. page 91–110, 2004.

[25] Garrick Orchard, Ajinkya Jayawant, Gregory K. Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in Neuroscience*, 9:437, 2015.

[26] Andrew J. Davison Patrick Bardow and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. *In IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, page 884–892, 2016.

[27] Jun Haeng Lee Byungkon Kang-Chang-Woo Shin Jooyeon Woo Jun-Seok Kim Yunjae Suh Sungho Kim Saber Moradi Ogan Gurel Paul K. J. Park, Kyoobin Lee and Hyunsurk Ryu. Computationally efficient, real-time motion recognition based on bio-inspired visual and cognitive processing. *In IEEE International Conference on Image Processing (ICIP). IEEE*, Sept. 2015.

[28] Jonathan Binas Matthew Cook Shih-Chii Liu Peter U. Diehl, Daniel Neil and Michael Pfeiffer. Fast-classifying, high accuracy spiking deep networks through weight and threshold balancing. *In Int. Joint Conf. Neural Netw. (IJCNN)*, 4:933–2940, 2015.

[29] Rubinstein R., Peleg T., and M. Elad. Analysis k-svd: A dictionary-learning algorithm for the analysis sparse model. *Signal Processing, IEEE Transactions*, 61(3):661–677, January 2013.

[30] B. Ramesh, H. Yang, G. M. Orchard, N. A. Le Thi, S. Zhang, and C. Xiang. Dart: Distribution aware retinal transform for event-based cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.

[31] Saiprasad Ravishankar and Yoram Bresler. Closed-form solutions within sparsifying transform learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, pages 5378–5382. IEEE, 2013.

[32] Saiprasad Ravishankar and Yoram Bresler. Doubly sparse transform learning with convergence guarantees. In *IEEE ICASSP 2014, Florence, Italy, May 4-9, 2014*, pages 5262–5266, 2014.

[33] Saiprasad Ravishankar and Yoram Bresler. ℓ_0 sparsifying transform learning with efficient optimal updates and convergence guarantees. *CoRR*, abs/1501.02859, 2015.

[34] Ryad Benosman Sio-Hoi Ieng Paul Rogister and Christoph Posch. Asynchronous event-based hebbian epipolar geometry. *IEEE Trans. Neural Netw.*, page 22(11):1723–1734, 2011.

[35] Ron Rubinstein, Alfred M. Bruckstein, and Michael Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.

[36] Ron Rubinstein, Tomer Peleg, and Michael Elad. Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model. *IEEE Trans. Signal Processing*, 61(3):661–677, 2013.

[37] Charles Clercq Chiara Bartolozzi Ryad Benosman, Sio-Hoi Ieng and Mandyam Srinivasan. Asynchronous frameless event-based optical flow. *Neural Netw.*, page 27:32–37, 2012.

[38] Josef Sivic and Andrew Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, page 591–606, Apr. 2009.

[39] Margarita Chli Stefan Leutenegger and Roland Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. *In Int. Conf. Comput. Vis. (ICCV)*, page 2548–2555, Nov. 2011. 2.

[40] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[41] A. Tsitiridis, C. Conde, I. M. de Diego, J. S. del Rio Saez, J. R. Gomez, and E. Cabello. Gabor feature processing in spiking neural networks from retina-inspired data. In *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015.

[42] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[43] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vis.*, page 57(2):137–154, May 2004.