

# Unsupervised Features Extraction from Asynchronous Silicon Retina through Spike-Timing-Dependent Plasticity

Olivier Bichler, Damien Querlioz, Simon J. Thorpe, Jean-Philippe Bourgoin and Christian Gamrat

**Abstract**—In this paper, we present a novel approach to extract complex and overlapping temporally correlated features directly from spike-based dynamic vision sensors. A spiking neural network capable of performing multilayer unsupervised learning through Spike-Timing-Dependent Plasticity is introduced. It shows exceptional performances at detecting cars passing on a freeway recorded with a dynamic vision sensor, after only 10 minutes of fully unsupervised learning. Our methodology is thoroughly explained and first applied to a simpler example of ball trajectory learning. Two unsupervised learning strategies are investigated for advanced features learning. Robustness of our network to synaptic and neuron variability is assessed and virtual immunity to noise and jitter is demonstrated.

## I. INTRODUCTION

THE overwhelming majority of vision sensors and processing systems currently in use are frame-based, where each frame is generally passed through the entire processing chain. Now for many applications, especially those involving motion processing, successive frames contain vast amounts of redundant information, which still need to be processed. This can have a high cost, in terms of computational power, time and energy. For motion analysis, local changes at the pixel level and their timing is really the only information one needs, and it may represent only a small fraction of all the data transmitted by a conventional vision sensor of the same sensibility.

Spiking silicon retinas, which are directly inspired from the way biological retinas work, are a direct response to the problematic exposed above. Instead of sending frames, silicon retinas use Address-Event Representation (AER) to asynchronously transmit spikes in response to local change in temporal and/or spatial contrast [1] [2]. In these devices, also called AER dynamic vision sensors, the addresses of the spikes are transmitted asynchronously (in real time) through a single serial link. Although relatively new, several types of spiking silicon retinas have already been successfully built, yet still with limited resolution of typically 128x128 pixels or less.

O. Bichler and C. Gamrat are with the CEA, LIST, Embedded Computers Laboratory, 91191 Gif-sur-Yvette Cedex, France (corresponding authors; phone: (+33)1.69.08.14.52; fax: (+33)1.69.08.83.95; e-mail: olivier.bichler@cea.fr, christian.gamrat@cea.fr).

D. Querlioz is with the Institut d'Electronique Fondamentale, Univ. Paris-Sud, CNRS, 91405, Orsay, France (e-mail: damien.querlioz@u-psud.fr).

S. J. Thorpe is with the CNRS Université Toulouse 3, Centre de Recherche Cerveau & Cognition, Toulouse, France (e-mail: simon.thorpe@cerco-ups-tlse.fr).

J.-P. Bourgoin is with the CEA, IRAMIS, Condensed Matter Physics Laboratory, 91191 Gif-sur-Yvette Cedex, France (e-mail: jean-philippe.bourgoin@cea.fr).

However, the undeniable advantages of silicon retinas are also what makes them more difficult to use, because most of the classic vision processing algorithms are inefficient or simply do not work with them [3]. Classical image-based convolutions for example are difficult to implement, because pixels activity is asynchronous and the AER data stream is continuous. Spike- or AER-based convolutional networks do exist [4], however the weights of the convolution kernel are often learned off-line and using a frame-based architecture. More importantly, those approaches are essentially based on the absolute spike rate of each pixel, thus ignoring much of the information contained in the relative timing between individual spikes [5].

To overcome these difficulties, we propose a novel approach that fully embraces the asynchronous and spiking nature of these sensors and is able to extract complex and overlapping temporally correlated features in a robust and completely unsupervised way. We show a new way of using Spike-Timing-Dependent Plasticity (STDP) to process true dynamic spike-based stimuli, recorded from an actual AER sensor, with what we hope will become a standard test case for such algorithms. We show how motion sequences of individual objects can be learned from complex moving sequences with a feed-forward multilayer unsupervised learning spiking neural network. This work, which extends some of the concepts introduced in [6], takes full benefit of the relative spike timing of the sensor's pixels and shows exceptional performances, considering the simplicity and the unsupervised nature of the proposed learning scheme. These characteristics also make this approach an excellent candidate for efficient future hardware implementations, that could take advantage of recent developments in memristive nano-devices [7].

## II. METHODOLOGY

In this paper we simulate a spiking neural network that performs pattern recognition based on AER retina data. To this end, a special purpose C++ event-based simulator was developed and is used for all the simulations. Event-based simulation is particularly well adapted for processing AER data flow, unlike traditional clock-driven neural network simulators, which generally focus more on biological modeling accuracy than efficient hardware simulation. Our simulator is therefore capable of processing 128x128 AER retina data in near real-time on a standard desktop CPU.

### A. Learning Rule

The learning rule, common to all the simulations presented in this paper, is a simplified STDP rule. STDP was demonstrated in biological neurons about a decade ago in [8] [9], and is now believed to be a foundation of learning of the brain [10] and is widely used, though with many variations, in both computational neuroscience [11] [12] and machine learning [13] [14] [15]. In our case, we use a simple rule where all the synapses of a neuron are equally depressed upon receiving a post-synaptic spike, except for the synapses that were activated with a pre-synaptic spike a short time before, which are strongly potentiated. It is important to note that all the other synapses are systematically depressed, even if they were never activated. This behavior therefore cannot be entirely modeled with a classical STDP window function  $\Delta w = f(t_{post} - t_{pre})$ . It is also not accurate to consider the synapses as being leaky, or volatile, because they only undergo Long-Term Depression (LTD) when the neuron is activated. If the neuron never fires, the weight of the synapses remains constant. The implications of this learning scheme are thoroughly discussed in the next section.

The general form of the weight update equations in the Long-Term Potentiation (LTP) case is the following:

$$\Delta w_+ = \alpha_+ \cdot \exp \left( -\beta_+ \cdot \frac{w - w_{min}}{w_{max} - w_{min}} \right) \quad (1)$$

In the LTD case, the equation is quite similar:

$$\Delta w_- = \alpha_- \cdot \exp \left( -\beta_- \cdot \frac{w_{max} - w}{w_{max} - w_{min}} \right) \quad (2)$$

where  $\alpha_+ > 0$ ,  $\beta_+ \geq 0$ ,  $\alpha_- < 0$  and  $\beta_- \geq 0$  are four parameters.  $w$  is the weight of the synapse and is allowed to change between  $w_{min}$  and  $w_{max}$ . Depending on the two  $\beta$  parameters, one can have either an additive ( $\beta = 0$ ) or a pseudo-multiplicative weight update rule, which can model different possible hardware (or software) implementations without compromising the working principle of the proposed scheme.

### B. Spiking Neuron Model

In our event-driven simulator, a spike event at time  $t_{spike}$  is modeled as the unit impulse function  $\delta(t - t_{spike})$ . Between two spikes, the integration  $u$  of the leaky integrate-and-fire neuron is the solution of the simple differential equation

$$u + \tau_{leak} \cdot \frac{du}{dt} = 0 \quad (3)$$

The neuron's integration state only needs to be updated at the next spike event, at time  $t_{spike}$ , where the synaptic weight  $w$  of the incoming spike is added to the integration :

$$u = u \cdot \exp \left( -\frac{t_{spike} - t_{last\_spike}}{\tau_{leak}} \right) + w \quad (4)$$

When the integration  $u$  reaches the neuron's threshold, a new spike event is created and sent to every output synapses. The integration is then reseted to zero and cannot increase again until the end of a refractory period  $T_{refrac}$ .

### C. Lateral Inhibition

When a neuron spikes, it disables all the other neurons during a period  $T_{inhibit}$ , during which no incoming spike is integrated. This inhibiting period also adds to the refractory period of the neurons recently activated, in the case where  $T_{inhibit} < T_{refrac}$ . Because the neurons are leaky, if  $T_{inhibit} \gg T_{leak}$ , one can consider that the neurons are also reset after the lateral inhibition.

### D. AER data

The AER data used in this paper were either recorded with the TMPDIFF128 DVS sensor [1] and downloaded from this website [16] or generated with the same format. An AER dataset simply consist of a list of events, with for each event, the address of the emitting pixel of the retina, the time-stamp of the event and its type. For the TMPDIFF128 sensor, a pixel generates an event each time the relative change of its illumination intensity reaches a positive or a negative threshold. Therefore, depending on the sign of the intensity change, events can be of either type ON or type OFF, corresponding to a increase or a decrease in pixel illumination, respectively.

## III. EXPERIMENTS AND RESULTS

In this section, we first present a simple learning case of short ball trajectories with artificially created AER data sequences, before moving to a real-life learning demonstration with a recorded sequence from a 128x128 AER silicon retina. Finally, we show the robustness of our approach, to external noise and jitter as well as to internal network parameters such as the weight evolution parameters and the neurons parameters.

### A. Partial Trajectory Extraction

For this first experiment, 8 computer generated AER data sequences were created, each representing a ball trajectory in a different direction, as shown figure 1. The characteristics of the generated data are identical to actual data recorded with the TMPDIFF128 sensor, with a lower resolution of 16x16 pixels. Every input pixel requires two synapses, to send the ON- and OFF-type events respectively, which makes a total  $2 * 16 * 16 = 512$  input addresses. Our neural network is constituted of 48 output neurons, with 512 synapses per neurons (see figure 2), each synapse being addressed by its corresponding event. Lateral inhibition is also implemented

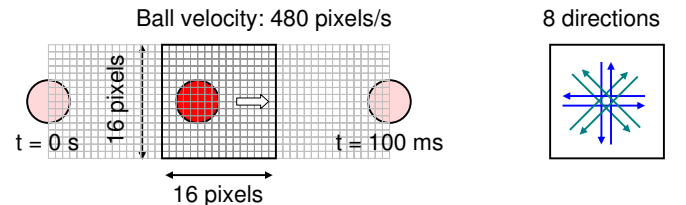


Fig. 1. Characteristics of the computer generated ball trajectories used as input stimuli of the network. A ball is moving in one of 8 directions at a 480 pixels/s velocity on a 16x16 pixels grid. AER events are generated by mimicking the properties of a spiking silicon retina.

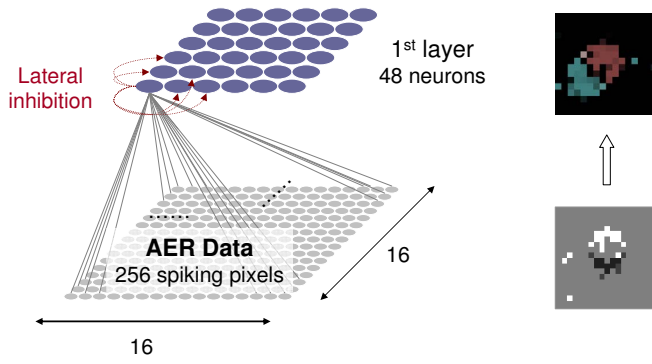


Fig. 2. Neural network topological overview for partial trajectory extraction. It is a one-layer feedforward fully connected network, with complete lateral inhibition, from each neuron to every other neuron. There is no spatially specific inhibition between neurons.

and each neuron inhibits the integration of all the other neurons during a time  $T_{inhibit}$  when it spikes.

When a neuron spikes at time  $t_{spike}$ , it potentiates the synapses that where the most recently activated, from  $t_{spike} - T_{LTP}$  to  $t_{spike}$ , and depresses all its other synapses. This increases the sensitivity of the neuron to the specific pattern that activated it, making it more likely to spike for a similar, correlated pattern, in the future. Because the neuron is leaky, only the contribution of sequences of spikes activating a majority of strongly potentiated synapses in a short time has a significant chance to rise the neuron's integration above the threshold. This ensure that the neuron is only sensitive to a specific pattern, typically a cluster of spikes strongly temporally correlated. The figures 3 and 4 show the activity of the network before and after the learning respectively.

Two mechanisms allow competitive and complementary learning of neurons [17]. The first one is lateral inhibition, which is fundamental to enable multiple neurons to learn

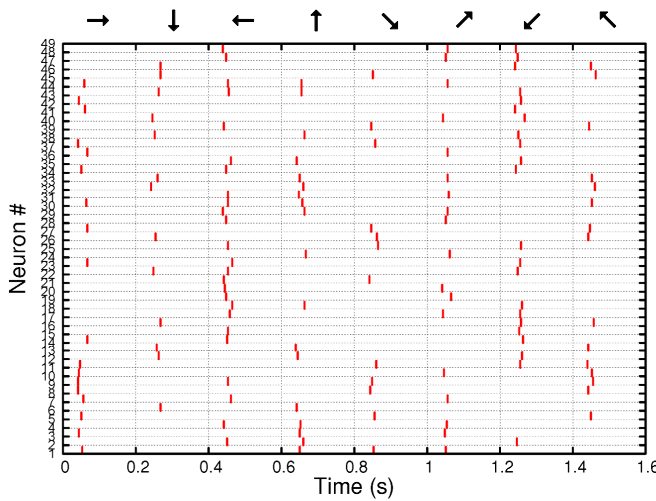


Fig. 3. Spiking events emitted by the output neurons (vertical axis) as a function of time (horizontal axis). The direction of the moving ball presented is indicated at the top of the plot. Initially, the weight of the synapses is on average equal to 80% of the maximum weight. The neurons are therefore very responsive, with no selectivity to the different trajectories, as can be seen when the 8 AER stimuli are presented in order, one every 200 ms.

multiple patterns. Without lateral inhibition, all the neurons would end up learning the same pattern. The inhibition time  $T_{inhibit}$  actually controls the minimum time interval between the chunks a trajectory can be decomposed into, each chunk being learned by a different neuron, as seen in figure 5. The second mechanism is the refractory period of the neuron itself, which contributes with the lateral inhibition to adapt the learning dynamic (in how many chunk the trajectory should be decomposed) to the input stimuli dynamic (how fast the motion is). If for example the motion is slow compared to the inhibition time, the refractory period of the neurons ensures that a single neuron cannot track an entire trajectory by repetitively firing and adjusting its weights to the slowly evolving input stimuli. Such a neuron would be “greedy”, as it would continuously send burst of spikes in response to various trajectories, when the other neurons would never have a chance to fire and learn something useful. After the learning, one can disable the lateral inhibition to verify that the neurons are indeed selective enough to be only sensitive to the learned pattern, as deduced from the weights reconstruction. From this point, even with continued stimuli presentation with STDP still enabled, the state of most of the neurons remains stable without lateral inhibition. A few of them adapt and switch to another pattern, which is perfectly fine since STDP is still in action. And more importantly, no “greedy” neuron appears.

The neuronal parameters for this simulation are summarized in table I. In general, the parameters for the synaptic weights are not critical for the proposed scheme (see table II). Only two important conditions should be ensured:

- 1) In all our simulations,  $\Delta w_+$  needed to be higher than  $\Delta w_-$ . In the earlier stage of the learning, the net effect of LTD is initially a lot stronger than LTP. Neurons are not selective and therefore all their synaptic weights are depressed on average. However, because the initial weights are randomly distributed and thanks to lat-

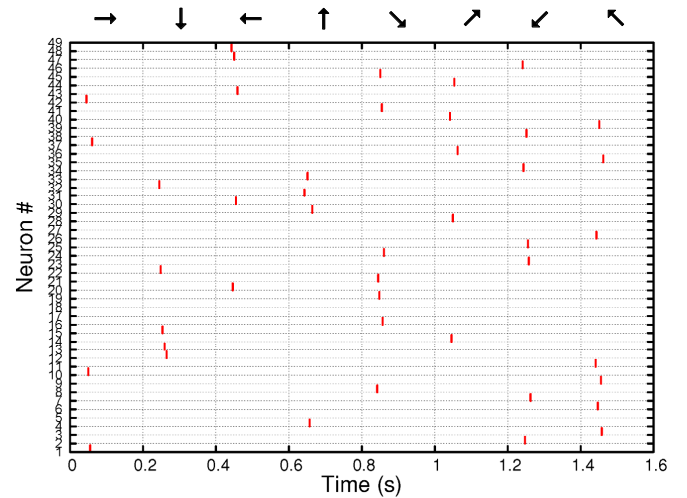


Fig. 4. After 2,000 presentations in random order, the 8 AER stimuli are again presented, one every 200 ms. Now, each neuron only responds to one particular part of one trajectory.

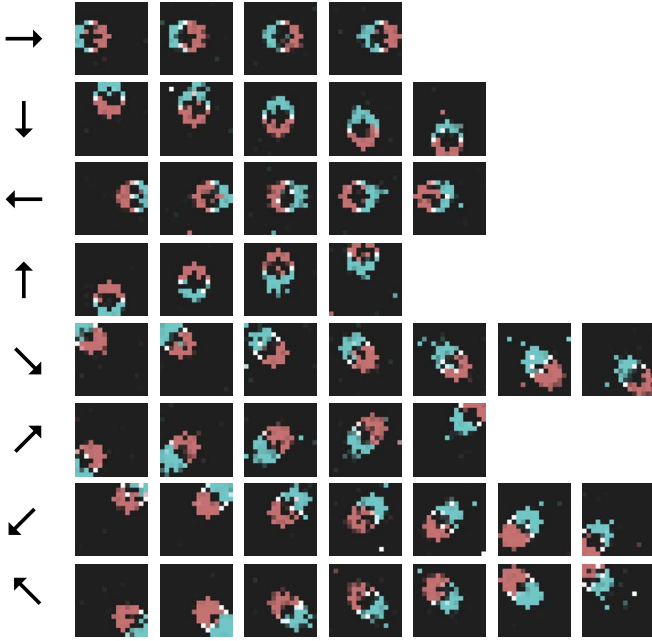


Fig. 5. Weight reconstructions after the learning. The reconstructions are ordered from the earliest neuron activated to the last one for each trajectory, according to the activity recording of figure 4. Red represents potentiated synapses linked to the positive (ON) output of the pixel and blue represents potentiated synapses linked to the negative (OFF) output of the pixel. When both ON and OFF synapses are potentiated for a given pixel, the resulting color is light-gray.

TABLE I  
DESCRIPTION AND VALUE OF THE NEURONS PARAMETERS FOR PARTIAL TRAJECTORY EXTRACTION.

Parameter	Value	Effect
$I_{thres}$	40000	The threshold directly affect the selectivity of the neurons. The maximum value of the threshold is limited by $T_{LTP}$ and $\tau_{leak}$ .
$T_{LTP}$	2 ms	The size of the temporal cluster to learn with a single neuron.
$T_{refrac}$	10 ms	Should be higher than $T_{inhibit}$ , but lower than the typical time the pattern this neuron learned repeats.
$T_{inhibit}$	1.5 ms	Minimum time interval between the chunks a trajectory can be decomposed into.
$\tau_{leak}$	5 ms	The leak time constant should be a little higher than to the typical duration of the features to be learned.

eral inhibition, at a certain point neurons necessarily become more sensitive to some patterns than others. At this stage, the LTP issued by the preferred pattern must overcome the LTD of the others, which is not necessarily guaranteed if  $\Delta w_+$  is too low. Note that if  $\Delta w_+$  is too high, the initial predicate does not hold and the neurons cannot be depressed enough to become selective.

- 2) One should have  $\Delta w < (w_{max} - w_{min})$ , but high precision is not required. 4 to 5 bits per weight is enough, as  $\Delta w_+ = 2 \cdot \Delta w_- = (w_{max} - w_{min})/10$  in our simulations.

It is remarkable that on average, there are 1.4 more neurons

TABLE II  
MEAN AND STANDARD DEVIATION FOR THE SYNAPTIC PARAMETERS, FOR ALL THE SIMULATIONS IN THIS PAPER. THE PARAMETERS ARE RANDOMLY CHOSEN FOR EACH SYNAPSE AT THE BEGINNING OF THE SIMULATIONS, USING THE NORMAL DISTRIBUTION.

Parameter	Mean	Std. Dev.	Description
$w_{min}$	1	0.2	Minimum weight (normalized).
$w_{max}$	1000	200	Maximum weight.
$w_{init}$	800	160	Initial weight.
$\alpha_+$	100	20	Weight increment.
$\alpha_-$	50	10	Weight decrement.
$\beta_+$	0	0	Increment damping factor.
$\beta_-$	0	0	Decrement damping factor.

activated for diagonal trajectories than for horizontal and vertical ones. This number is consistent with the distance ratio between these two types of trajectory, which is equal to  $\sqrt{2}$ .

### B. Advanced Features Learning

In this section, we show how the learning scheme introduced above can be used to extract more complex, temporally overlapping features, directly from an AER silicon retina. The stimulus used in this section was recorded from the TMPDIFF128 DVS sensor by the group of T. Delbruck and is freely available on this website [16]. It represents cars passing under a bridge over the 210 freeway in Pasadena. The sequence is 78.5 s in duration, containing a total of 5.2M events, with an average event rate of 66.1k events per second. Figure 6 shows some rendering of the sequence obtained with the jAER software [18] that accumulates the events during a short period of time in order to draw an image. Counting the number of cars passing on each traffic lane by watching this sequence with the naked eye is actually almost an impossible task, because there are no landmarks to distinguish the lanes other than the moving cars and the traffic is quite dense.

The neural network used for this simulation is described figure 7. It is a two-layer feedforward fully connected network, with 60 neurons in the first layer and 10 neurons in the second one. The total number of synapses in this system is  $2 * 128 * 128 * 60 + 60 * 10 = 1,966,680$ , which could however be greatly reduced in practical applications where a fully connected network is generally not necessary. This would be the case of our example, because the size of the features that can be learned (the cars) is small compared

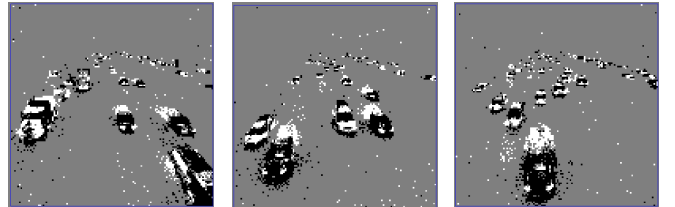


Fig. 6. Illustration of the dataset used for advanced features learning: cars passing under bridge over the 210 freeway in Pasadena. White pixels represents ON events and black pixels OFF events. This AER sequence and other ones are available online [16].



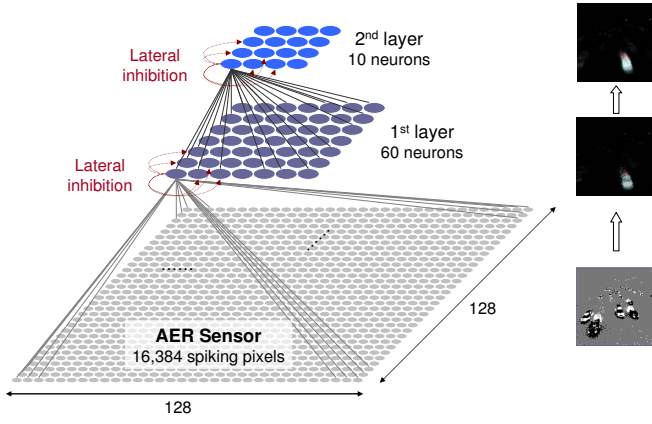


Fig. 7. Neural network topological overview for advanced features learning, directly from data recorded with the AER sensor. It is a two-layer feedforward fully connected network, with complete lateral inhibition, from each neuron to every other neuron. There is no spatially specific inhibition between neurons. The bottom layer is the AER sensor and is not considered as a layer of the neural network.

to the size of the retina and their spatial locations are also well defined. Nevertheless, we wanted to show the power of our approach by not spatially constraining the inputs of the neurons.

Two learning strategies are successively tested in the following, both completely unsupervised. The first one could be called a “global” learning, where the two layers learn concurrently, the lateral inhibition being always enabled. In the second strategy, only the first layer is active in a first step. Once the learned features are stable, lateral inhibition is removed and STDP can be disabled for this layer. Only after this step is the second layer allowed to learn and lateral inhibition is also removed afterwards. In this strategy, there is no more lateral inhibition involved in the network once every neuron has specialized itself and we will show the advantages of this method to achieve exhaustive extraction of temporally overlapping features. Finally, a methodology to find the optimal neuronal parameters through genetic evolution algorithm is detailed.

1) *Global Learning*: In this first learning strategy, the two neuronal layers learn at the same time and the lateral inhibition is always enabled. If one considers only the first layer, this experiment is exactly the same as the previous one with the ball trajectories. It is remarkable that although the cars trajectories constantly overlap in time, the traffic being quite dense, the mechanism described earlier still successfully extracts trajectories associated with a single traffic lane, as demonstrated with the weight reconstruction of the neurons of the first layer shown in figure 8. Because there is no particular correlation between the cars in different lanes, two groups of synapses spatially belonging to different traffic lanes cannot in average be potentiated together. Thanks to initial conditions and lateral inhibition, the neuron necessarily become more sensitive to one of the two groups, thus allowing LTP to potentiate one group more, regardless of the other synapses activated at the same time, which will on average undergo LTD because they are not correlated

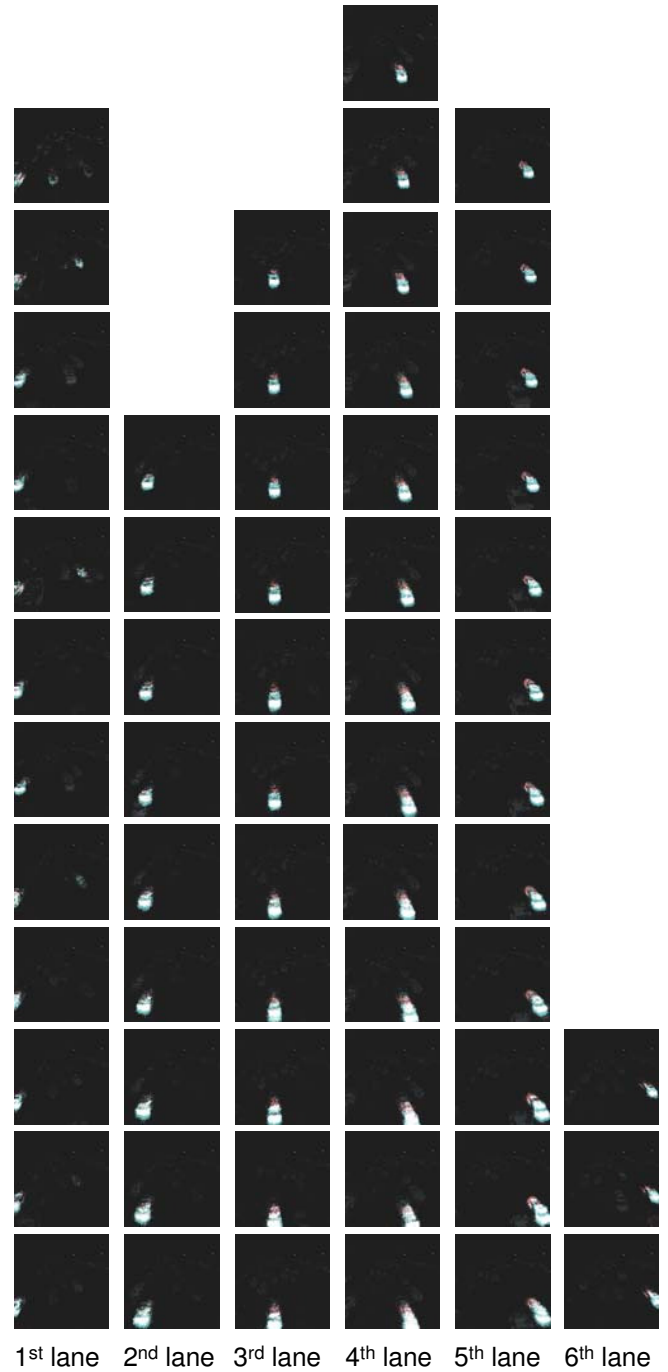


Fig. 8. Weight reconstructions of the first neuronal layer after the learning of the cars sequence. There are 60 neurons and each of them is sensitive to a specific part of the trajectory for only one traffic lane. Red represents potentiated synapses linked to the positive (ON) output of the pixel and blue represents potentiated synapses linked to the negative (OFF) output of the pixel. When both ON and OFF synapses are potentiated for a given pixel, the resulting color is light-gray.

temporally. If the threshold is sufficiently high to allow a good selectivity of the neuron, cars activating this group of synapses will eventually be sufficient to make it fire most of the time. This only works if LTD is systematically applied to synapses not undergoing an LTP, even those not receiving a pre-synaptic spike. Therefore, classical STDP mechanisms

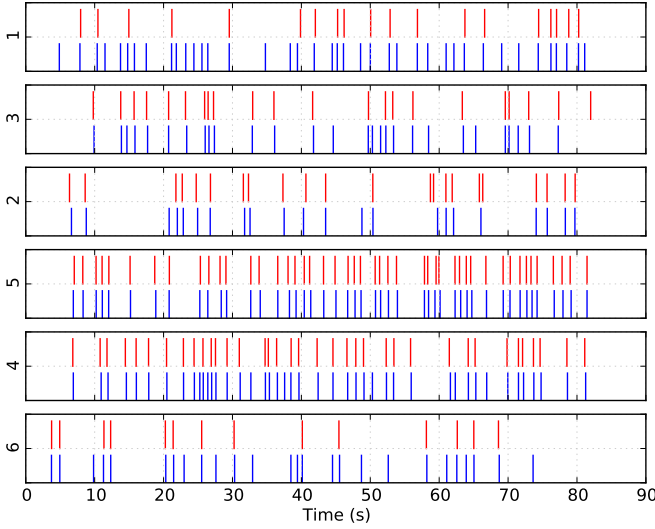


Fig. 9. Detection of the cars on each traffic lane after the learning, with the global strategy. The reference activity, obtained by hand labeling, is compared to the activity of the best neuron of the second layer for the corresponding traffic lane (numbered from 1 to 6). The reference activity is at the bottom of each subplot (in blue) and the networks output activity is at the top (in red).

modeled by the equation  $\Delta w = f(t_{post} - t_{pre})$  fail at this task, because it is not possible with this simple rule to depress synapses whose activation time is precisely not correlated with the post-synaptic spike.

Using the same mechanism, a second neuron layer fully connected to the first one is able to perform advanced features learning from the partial trajectories extracted with the first layer. With appropriate parameters (see table III), this second layer can identify entire traffic lanes by recombining partial trajectories. The output activity of this layer can be used to partially count the number of cars passing on each traffic lane, as shown by the activity raster plot in figure 9. The detection rate ranges from 47% for the first lane to 100% for the fifth lane.

The activity raster plot and weight reconstructions are computed after the input AER sequence of 78.5 s has been presented 8 times. This corresponds to a real-time learning duration of approximatively 10 minutes, after which the evolution of the synaptic weights stay very weak. It is

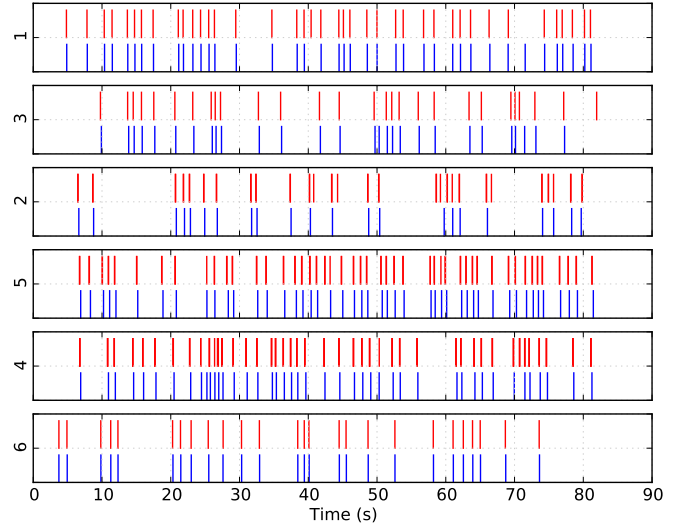


Fig. 10. Detection of the cars on each traffic lane after the learning, with the optimized, layer-by-layer strategy. The reference activity, obtained by hand labeling (shown in blue), is compared to the activity of the best neuron of the second layer for the corresponding traffic lane (numbered from 1 to 6) - shown in red.

notable that even after only one presentation of the sequence, the beginning of the specialization of most of the neurons is already apparent from the weight reconstructions and a majority of the visible extracted features at this stage remains stable until the end of the learning.

2) *Layer-by-layer Learning*: As we showed with the learning of partial ball trajectories, lateral inhibition is no longer necessary when the neurons are specialized. In fact, lateral inhibition is not even desired, as it could prevent legitimate neurons from firing in response to temporally overlapping features. This does not prevent the learning in any case provided the learning sequence is long enough to consider that the features to learn are temporally uncorrelated. This mechanism, which is fundamental to allow competitive learning, therefore leads to poor performances in terms of pattern detection once the learning become stable. In conclusion, the more selective a neuron is, the less it needs to be inhibited by its neighbors.

The figure 10 shows the activity of the output neurons of the network when lateral inhibition and STDP are disabled after the learning, on the first layer first, then on the second layer. The weight reconstructions for the second layer are also shown in figure 11. For each neuron of the second layer, the weight reconstruction is obtained by computing the weighted sum of the reconstructions of the first layer, with the corresponding synaptic weights for each neuron of the second layer. The real-time learning duration is 10 minutes per layer, that is 20 minutes in total. Now that neurons cannot be inhibited when responding to their preferred stimuli, near exhaustive features detection is achieved. The network really learns to detect cars passing on each traffic lane in a completely unsupervised way, with only 10 tunable parameters for the neurons in all and without having ever programmed the neural network to do so. We are able to count the cars

TABLE III  
NEURONS PARAMETERS FOR ADVANCED FEATURES LEARNING. A DIFFERENT SET OF PARAMETERS IS USED DEPENDING ON THE LEARNING STRATEGY (GLOBAL OR LAYER-BY-LAYER).

Parameter	Global Learning		Layer-by-layer Learning	
	1st Layer	2nd Layer	1st Layer	2nd Layer
$I_{thres}$	500000	1500	1060000	2240
$T_{LTP}$	12 ms	300 ms	14.7 ms	46.5 ms
$T_{refrac}$	300 ms	250 ms	517 ms	470 ms
$T_{inhibit}$	50 ms	100 ms	10.2 ms	182 ms
$\tau_{leak}$	450 ms	300 ms	187 ms	477 ms
Recog. rate	47% to 100% / lane		<b>98%</b> overall	

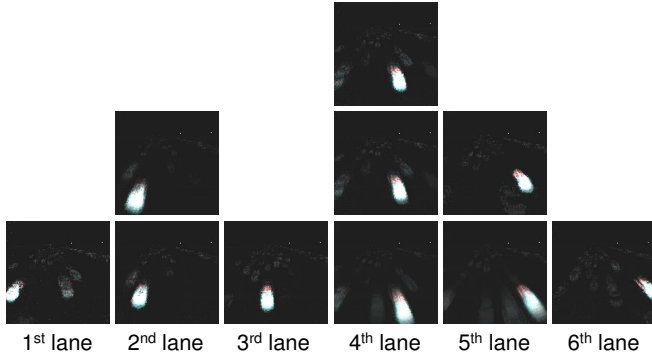


Fig. 11. Weight reconstructions for the second layer after the learning with the layer-by-layer strategy (obtained by computing the weighted sum of the reconstructions of the first layer, with the corresponding synaptic weights for each neuron of the second layer). The neurons of the second layer associate multiple neurons of the first layer responding to very close successive trajectory parts to achieve robust detection in a totally unsupervised way.

passing on each lane at the output of the network with a fairly good accuracy simply because this is the consequence of extracting temporally correlated features. Over the 207 cars passing on the six lane during the 78.5 s sequence, only 4 cars are missed, with a total of 9 false positives, corresponding essentially to trucks activating neurons twice or cars driving in the middle of two lanes, which were not specifically labeled. This gives an impressive detection rate of 98% even though no fine tuning of parameters is required.

If lateral inhibition is removed after the learning, but STDP is still active, we observed that the main features extracted from the first layer remain stable, as it was the case for the ball trajectories learning.

3) *Genetic Evolution*: Finding optimal values for the neuron's parameters  $I_{thres}$ ,  $T_{LTP}$ ,  $T_{refrac}$ ,  $T_{inhibit}$  and  $\tau_{leak}$  can be a challenging task. However, since all the neurons in a same layer share the same parameters, this makes only 10 different parameters in total in this neural network that must be fitted to a particular type of stimuli. This task can be accomplished efficiently by using a genetic algorithm, provided that a target network activity can be properly formulated. Multiple instances of the neural network with randomly mutated parameters are allowed to learn in parallel and a score for each instance is computed at the end of the learning. The parameters of the instances with the best scores are mutated again for the next run. The score is calculated by comparing the activity of the second layer and the reference activity obtained by hand labeling. The activity spike trains are convolved with the Gaussian function  $\exp(-(t/\tau)^2)$  to form a continuous signal. The absolute value of the difference of the resulting signals, for the output activity of the network and the reference activity, is then integrated and normalized. Decent parameters can be found in less than 10 generations, with 80 runs and 8 winners per generation.

### C. Robustness and Noise Immunity

In this section, we show that our learning scheme is remarkably tolerant to synaptic variability, even when neurons variability is added as well. Exact, matched numbers for the

TABLE IV  
DETECTION RATE STATISTICS OVER 100 SIMULATIONS, WITH A DISPERSION OF 20% FOR ALL THE SYNAPSES PARAMETERS. THE DISPERSION IS DEFINED AS STANDARD VARIATION OF THE MEAN VALUE.

Lanes learned	Missed cars <sup>†</sup>	False positives <sup>†</sup>	Total (%)
First five	$\leq 10$	$\leq 10$	79
	$> 10$ and $\leq 20$	$\leq 10$	10
	$> 20$	$\leq 10$	2
Only four	$\leq 10$	$\leq 10$	9
<sup>†</sup> On learned lanes			<b>100</b>

neurons parameters are therefore not required. This shows that the network is robust and does not require fine tuning of its parameters to work properly. We also show extremely strong tolerance to noise and jitter, in levels far superior to the already noisy data recorded from the AER sensor.

1) *Synaptic variability*: We performed a basic analysis of the robustness to synaptic variability for our specific learning example. The table IV summarizes the results in terms of missed cars and false positives for a batch of 100 simulations, where a dispersion of 20% is applied to all the synapse's parameters:  $w_{min}$ ,  $w_{max}$ ,  $w_{init}$ ,  $\alpha_+$  and  $\alpha_-$  ( $\beta_+ = \beta_- = 0$ ). This is a considerable amount of variability: 20% of the synapses have a maximum weight that is 25% higher or lower than the average value. Over the 100 simulations, 9 failed to learn more than 4 traffic lanes, but even when two traffic lanes are not learned, the detection rate for the others remains better than 95%. The sixth traffic lane is never learned. This is actually understandable, because cars passing on the sixth traffic lane (at the very right of the retina) activated less pixels over their trajectory than those on other lanes, with a total amount of cars that is also lower. Consequently, because the overall spiking activity for lane 6 is at least 50% lower than the others, it is likely that depending on the initial conditions or some critical value for some parameters, no neuron is able to sufficiently potentiate the corresponding synapses to gain exclusive selectivity. Indeed, figure 8 shows a specific example where all the lanes are learned and only 3 neurons out of 60 manage to become sensitive to the last lane.

2) *Neuronal variability*: A new batch of 100 simulations was performed, this time with an added dispersion of 10%

TABLE V  
DETECTION RATE STATISTICS OVER 100 SIMULATIONS WITH A DISPERSION OF 10% FOR ALL THE NEURONS PARAMETERS, IN ADDITION TO THE DISPERSION OF 20% FOR ALL THE SYNAPSES PARAMETERS.

Lanes learned	Missed cars <sup>†</sup>	False positives <sup>†</sup>	Total (%)
All six	$\leq 20$	$\leq 10$	1
First five	$\leq 10$	$\leq 10$	51
	$> 10$ and $\leq 20$	$\leq 10$	21
	$> 20$	$\leq 10$	5
Five (others)	$\leq 10$	$\leq 10$	3
Only four	$\leq 10$	$\leq 10$	16
	$> 10$ and $\leq 20$	$\leq 10$	1
	$> 20$	$\leq 10$	2
<sup>†</sup> On learned lanes			<b>100</b>

applied to all the neuronal parameters from table III, for the two layers. Results in table V are still good for 75% of the runs and very good for about 50% of them, if one ignores the sixth lane, which is very hard to learn in a completely unsupervised way. In the worst cases, only four lanes are learned. It is noteworthy that the lanes 4 and 5 are always correctly learned, with constantly more than 90% of detected cars in all the simulations. The reason is that these lanes are well identifiable (contrary to lanes 1 and 6) and experience the highest traffic (the double compared to lanes 2 and 3). It is likely that with a longer AER sequence, better results could be achievable, without even considering the possibility of increasing the resolution of the sensor.

3) *Noise and Jitter*: The robustness to noise and jitter of the proposed learning scheme was also investigated. Simulation with added white noise (such that 50% of the total amount of the spikes in the sequence are random) and 5 ms added random jitter shown almost no impact on the learning at all. Although only the first five traffic lanes are learned, essentially for the reasons exposed above, there were less than 5 missed cars and 10 false positives with the parameters from table III.

#### IV. DISCUSSION AND CONCLUSION

This paper introduced the first practical unsupervised learning scheme capable of exhaustive extraction of temporally overlapping features directly from unfiltered AER silicon retina data, using only a simple, fully local STDP rule and 10 parameters in all for the neurons. We showed how this type of spiking neural network can learn after only 10 minutes of real-life data to detect cars with an accuracy greater than 95%, with a limited retina size of only 128x128 pixels.

The next logical step to improve our learning scheme would be to implement a more progressive deactivation of the lateral inhibition, which would take place during the learning. Neurons should be able to reduce the strength of the lateral inhibition proportionally to their selectivity. The only difficulty is to reliably quantize the selectivity of the neurons during the learning, without too much overhead.

Such a neural network could very well be used as a pre-processing layer for an intelligent motion sensor, where the extracted features could be automatically labeled and higher-level object tracking could be performed for example. The STDP learning rule being very loosely constrained and fully local, no complex global control circuit would be required. This also paves the way to very efficient hardware implementations that could use large crossbars of memristive nano-devices.

#### REFERENCES

- [1] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128x128 120 db 15 s latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.
- [2] K. A. Zaghloul and K. Boahen, "Optic nerve signals in a neuromorphic chip: Part I and II," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 4, pp. 657–675, 2004.
- [3] T. Delbruck, "Frame-free dynamic digital vision," in *Intl. Symp. on Secure-Life Electronics, Advanced Electronics for Quality Life and Society*, 2008, pp. 21–26.
- [4] J. Pérez-Carrasco, C. Serrano, B. Acha, T. Serrano-Gotarredona, and B. Linares-Barranco, "Spike-based convolutional network for real-time processing," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, 2010, pp. 3085–3088.
- [5] R. Guyonneau, R. VanRullen, and S. J. Thorpe, "Temporal codes and sparse representations: A key to understanding rapid processing in the visual system," *Journal of Physiology-Paris*, vol. 98, no. 4-6, pp. 487–497, 2004.
- [6] T. Masquelier, R. Guyonneau, and S. J. Thorpe, "Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains," *PLoS ONE*, vol. 3, no. 1, p. e1377, 2008.
- [7] S. H. Jo, K.-H. Kim, and W. Lu, "High-density crossbar arrays based on a si memristive system," *Nano Letters*, vol. 9, no. 2, pp. 870–874, 2009.
- [8] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, "Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs," *Science*, vol. 275, no. 5297, pp. 213–215, 1997.
- [9] G.-Q. Bi and M.-M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," *The Journal of Neuroscience*, vol. 18, no. 24, pp. 10 464–10 472, 1998.
- [10] Y. Dan and M.-M. Poo, "Spike timing-dependent plasticity of neural circuits," *Neuron*, vol. 44, no. 1, pp. 23–30, 2004.
- [11] S. M. Bohte and M. C. Mozer, "Reducing the variability of neural responses: A computational theory of spike-timing-dependent plasticity," *Neural Computation*, vol. 19, no. 2, pp. 371–403, 2007.
- [12] E. M. Izhikevich and N. S. Desai, "Relating STDP to BCM," *Neural Computation*, vol. 15, no. 7, pp. 1511–1523, 2003.
- [13] B. Nessler, M. Pfeiffer, and W. Maass, "STDP enables spiking neurons to detect hidden causes of their inputs," in *Advances in Neural Information Processing Systems*, vol. 22, 2010, pp. 1357–1365.
- [14] T. Masquelier and S. J. Thorpe, "Unsupervised learning of visual features through spike timing dependent plasticity," *PLoS Comput Biol*, vol. 3, no. 2, p. e31, 2007.
- [15] A. Gupta and L. Long, "Character recognition using spiking neural networks," in *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, 2007, pp. 53–58.
- [16] DVS128 Dynamic Vision Sensor Silicon Retina data. (2011, Jan.). [Online]. Available: <http://sourceforge.net/apps/trac/jaer/wiki/AER%20data>
- [17] T. Masquelier, R. Guyonneau, and S. J. Thorpe, "Competitive stdp-based spike pattern learning," *Neural Comput.*, vol. 21, pp. 1259–1276, 2009.
- [18] jAER Open Source Project. (2011, Jan.). [Online]. Available: <http://jaer.wiki.sourceforge.net>