2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Event-based Features for Robotic Vision

Xavier Lagorce, Sio-Hoi Ieng and Ryad Benosman

Abstract— **This paper introduces a new time oriented visual feature extraction method developed to take full advantage of an asynchronous event-based camera. Event-based asynchronous cameras encode visual information in an extremely optimal manner in term of redundancy reduction and energy consumption. These sensors open vast perspectives in the field of mobile robotics where responsiveness is one of the most important needed property. The presented technique, based on echo-state networks will be shown particularly suited for unsupervised features extraction in the context of high dynamic environments. Experimental results are presented, they show the method adequacy with the high data sparseness and temporal resolution of event-based acquisition. This allows features extraction at millisecond accuracy with a low computational cost.**

I. Introduction

Feature extraction is at the basis of almost every machine vision system. They represent valuable information coming from the environment. A feature is always sought as a spatial patch due to the current use of images as the structure to acquire and store light intensities reflected by objects in a scene. But images deal poorly in capturing the valuable temporal information present in natural scenes because they are static snapshots. Increasing the dynamics of the acquisition implies higher frame rates that then produce high amount of data. This approach is in general incompatible with embedded systems and is very limiting to high-level applications as it requires a lot of computational power which is problematic for embedded robotic tasks.

On the other hand, biological observations confirm that images are unknown to the visual system. Biological retinas encode visual data asynchronously as sparse spiking outputs rather than frames of pixels' values [1] which allows the whole perceptual system to be massively parallel and data-driven.

This paper introduces an unsupervised system that allows to extract visual spatiotemporal features from scenes. It does not rely on images but on the precise timing of spikes acquired by an asynchronous event-based silicon retina. The development of asynchronous event-based retinas has been pioneered by the work of Mahowald and Mead [2]. Neuromorphic asynchronous event-based retinas allow, as we will show, to derive new insights into the study of perceptual models and the introduction of time oriented visual features. Current available event-based vision sensors output compressed digital data in the form of events, reducing latency and increasing temporal range compared to conventional imagers. A complete review of the history and existing sensors can be found in [3].

The presented model is based on Echo-State Networks (ESN) which is a reservoir computing based technique of recurrent neural networks particularly suited for dynamical signal learning [4][5]. ESNs allow to preserve the dynamic of input signals and to extract complex temporal patterns. The primitives extraction from asynchronous event-based inputs combines the use of a set of Echo-State Networks and a Winner-Take-All (WTA) technique allowing automatic selection among the ESNs [6][7]. This selection introduced by the WTA network forces each ESN to learn different primitives based on movement prediction. The methods and algorithms presented in this paper have all been used on real data.

II. Event-based asynchronous sensors

The Dynamic Vision Sensor (DVS) used in this work is an Address-Event Representation (AER) silicon retina with 128×128 pixels [8] (Fig. 1(a)). The DVS output consists of asynchronous address-events that signal scene luminance changes at the times they occur. Each pixel is independent and detects changes in log intensity larger than a threshold since the last event it emitted (typically 15% contrast). When the change in log intensity exceeds a set threshold (see Fig. 1(b)), an ON or OFF event is generated by the pixel depending on whether the log intensity increased or decreased (see Fig. 1(c)). Since the DVS is not clocked like conventional cameras, the timing of events can be conveyed with a very accurate temporal resolution of approximately 1μs. Thus, the "effective frame rate" is typically several kHz. We can define an event occurring at time t at the pixel $[x, y]^T$ as :

$$e(x, y, t) = p \tag{1}$$

where p is the polarity of the event which can take the value $+1$ or -1 if the event encodes a change of the measured signal toward respectively higher or lower value. This data-driven process suppresses redundancy and the coding of the exact time of input signal's changes allows a very high dynamic of acquisition.

III. Feature extraction

A. General architecture

The idea developped in this work is to identify and select features for spiking retinas using an unsupervised and less redundant representation of temporal and spatial input patterns of asynchronous events. To achieve this, we use a set of independent Echo-state networks (ESNs) as

X. Lagorce, S-H. Ieng, R. Benosman are with the Institut de la Vision, University of Pierre and Marie Curie-UPMC/Inserm, France xavier.lagorce@etu.upmc.fr, sio-hoi.ieng@upmc.fr, ryad.benosman@upmc.fr

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

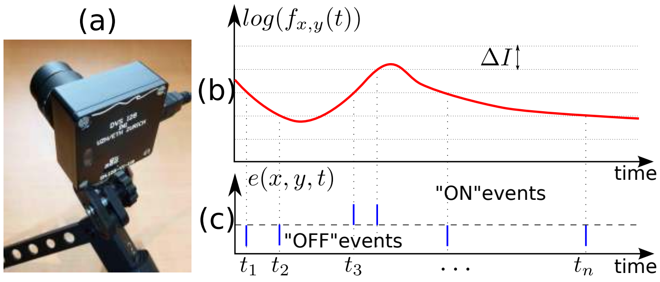


Fig. 1. 128 × 128 pixel array DVS (a), typical signal showing the log of luminance of a pixel located at $[x, y]^T$ (b) and asynchronous temporal contrast events generated by this pixel in response to this light variation (c).

predictors of future outputs. Fig. 2 provides the general scheme of the method. The output of the DVS retina is sent to each ESN after a conversion process. Each ESN will then represent an abstract spatiotemporal sequence of events and predict the content of the environment in the next step. The input neurons of each receptive field receive converted spikes streams from the DVS into analog signals in order to comply with the ESNs' input format. Each ESN is trained to predict the analog input signal one time step ahead. The last layer of the system implements a Winner-Take-All (WTA) neural network to select the best predictor among the set of ESNs. The WTA inhibits poorly predicting ESNs to ensure that the best predictor receives sufficient time to learn a particular spatiotemporal sequence. This selected ESN can then specialize in recognizing the spatiotemporal pattern and predicting its temporal evolution. The WTA mechanism ensures that each ESN focuses on an independent feature. The system thus enforces the rule that a same pattern can not be predicted by two different ESNs. Consequently, at a given time, the winning network in the WTA layer will indicate which feature is present. With each ESN being randomly initialized, the WTA layer makes the learning process completely unsupervised. The features are automatically extracted by the system from the input signal. The following subsections will describe in more details the different layers presented in Fig. 2.

B. Signal pre-processing

The DVS spiking retina used is of size 128×128, meaning that there are about $16K$ pixels in total. A direct use of reservoir computing on these pixels would lead to a network with $16K$ input neurons and 10 to 100 times more hidden neurons (a typical setup of the reservoir). To simulate and train such a huge network is certainly far beyond the reach of modern computers. Some preprocessing steps are then needed to reduce the input dimension.

After being resampled, the events output by the DVS are divided into several receptive fields $RF(x_0, y_0, t_1, t_2)$ which represent some spatio-temporal volumes of spikes defined by :

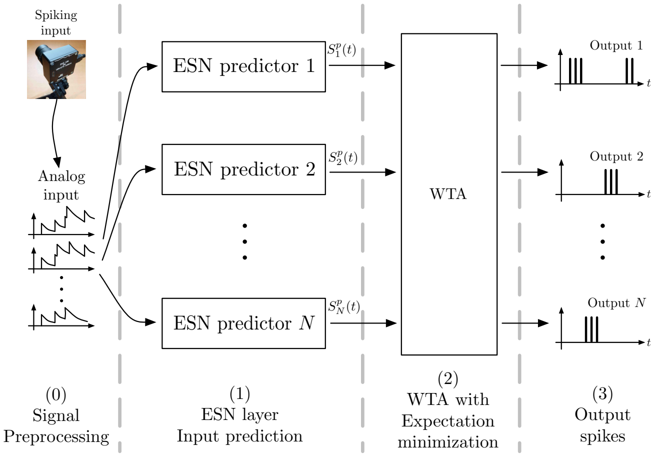


Fig. 2. Unsupervised feature extraction: spikes from the DVS are transformed into analog inputs that are sent to a set of Echo-state networks (ESNs). Each ESN is trained to predict future outputs based on current and passed activities. They output S_k^p, a representation of their prediction quality which is fed into a Winner take all network. This WTA selects the best predicting ESN and allows it to train on the sequence. The combined use of prediction and inhibition allows each ESN to specialize in the prediction of a particular feature and thus act as a feature detection.

$$\begin{aligned} RF(x_0, y_0, t_1, t_2) = & \\ \{e(x, y, t) \,|\, t \in [t_1, t_2], & \\ x \in [x_o - \Delta_x, x_o + \Delta_x], & \\ y \in [y_o - \Delta_y, y_o + \Delta_y]\} & \end{aligned} \tag{2}$$

Finally, we can consider a decay function $G(t, t_0)$:

$$G(t, t_0) = e^{-(t-t_0)/\tau} \tag{3}$$

which is applied to all the spikes emitted from the pixels of a receptive field. It provides an analog output signal A for each pixel (x, y) covered by $RF(x_0, y_0, t_0, t)$:

$$A(x, y, t_0, t) = \sum_{e(x,y,t_i) \in RF(x_0,y_0,t_0,t)} |e(x, y, t_i)| \cdot G(t, t_i) \tag{4}$$

This analog signal can then be used as an input for the ESNs.

The output of the input neurons layer is the vector constituted of all the $A(x_i, y_i, t_0, t)$ of the different pixels contained in $RF(x_0, y_0, t_0, t)$. In the following, for clarity, we will consider only one receptive field and will thus call this vector $\underline{A}(t)$:

$$\underline{A}(t) = \begin{pmatrix} A(x_i, y_i, t_0, t) \\ \vdots \\ A(x_M, y_M, t_0, t) \end{pmatrix} \tag{5}$$

C. ESN layer – Input prediction

This layer, marked as (1) in Fig. 2, refered to as the prediction layer, is made of N ESNs. An ESN k is defined by its internal state s^k, and some weight matrices W_{out}^k, W_{in}^k and W_{r}^k. These weight matrices are random and different for each ESN. They are used to compute the evolution of the internal state of the ESN and its output (out^k) with the following relations :

$$\begin{aligned} s^k(t_n) &= W_{\mathrm{r}} \cdot s^k(t_{n-1}) + W_{\mathrm{in}} \cdot \underline{A}(t_n) & (6) \\ out^k(t_n) &= W_{\mathrm{out}} \cdot s^k(t_n) & (7) \end{aligned}$$

Each ESN is trained to predict $\underline{A}(t_n)$ one time step ahead according to the relation (8), with k refering to the k^{th} ESN :

$$\underline{\hat{A}}_k(t_n + dt) = out^k(t_n) = W^k_{\mathrm{out}} \cdot s^k(t_n), \qquad (8)$$

The training process is based on an online recursive least squares method described in [9] using the prediction error vector $\epsilon^p_k(t_n)$:

$$\epsilon^p_k(t_n) = \underline{\hat{A}}_k(t_n) - \underline{A}(t_n), \qquad (9)$$

The output signal of the Prediction Layer (see Fig. 2), is the similarity measure $S^p_k(t_n)$ between the input signal $\underline{A}(t_n)$ and the prediction $\underline{\hat{A}}_k(t_n)$ provided by ESN k whose value inscreases with the quality of the prediction.

If we denote V_i the i^{th} component of the vector V, then $S^p_k(t_n)$ is :

$$S^p_k(t_n) = \frac{\sum_i \left| \underline{A}(t_n)_i \cdot \underline{\hat{A}}_k(t_n)_i \right|}{\sum_i |\underline{A}(t_n)_i| \cdot \sum_i \left| \underline{\hat{A}}(t_n)_i \right|}, \qquad (10)$$

where $\underline{A}(t_n)$ and $\underline{\hat{A}}(t_n)$ have been normalized between 0 and 1.

D. Winner-Take-All selection

The third layer of the model selects the best predictor among the N ESNs. A WTA network is constituted of the neurons $W_1 \ldots W_N$ and of an inhibitory neuron ([6][7]).

This WTA aims at selecting its input which shows the maximum similarity measure. The similarity measures $S^p_k(t_n)$ obtained from layer (1) are first transformed into spike trains via a first layer of non-leaky Integrate-and-Fire (IF) neurons. For the system to be more reliable, these neurons (see Fig. 3) do not directly integrate the value of S^P_k but a value transformed through a sigmoid-shaped gain g_{IF}:

$$g_{\mathrm{IF}}(x) = G_{\min} + \frac{G_{\max} - G_{\min}}{1 + \exp(-(x - x_0)/\lambda)}, \qquad (11)$$

where $G_{\min}$, $G_{\max}$, x_0 and λ are tunable parameters.

The values of $G_{\min}$ and $G_{\max}$ determine the minimum and maximum values of the firing rates output by the IF neurons and are set in the experiment for spike rates spanning from 5kHz to 15kHz. The value of λ determines the shape of the sigmoid. Higher values of this parameter allow the system to efficiently discrimine values closer to each others.

Finally, a regulator is introduced to dynamically set the value of x_0 close to the current value of S^p_k.

The output of the network is then the best predictor $W(t)$:

$$W(t) = \arg \max_{k=1..N} S^p_k(t). \qquad (12)$$

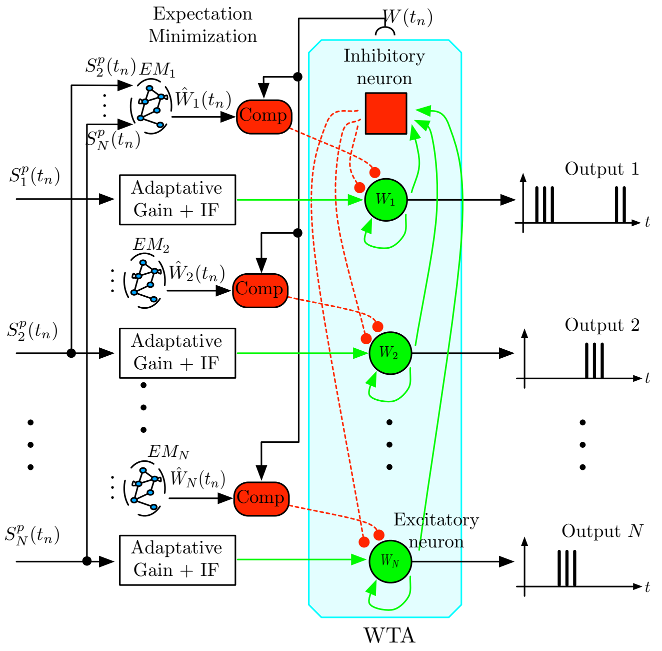


Fig. 3. The WTA network is composed of excitatory neurons $W_1 \ldots W_N$ and one inhibitory neuron. This network is fed by a first layer of non-leaky Integrate-and-Fire neurons which transforms the similarity measures $S^p_1 \ldots S^p_N$ into spike trains with an adaptive gain assuring a correct response of the WTA when S^p_k varies largely. The WTA network allows the selection of the best predictor $W(t_n)$ among the N ESNs of the previous layer. To prevent several networks to be selected by the same input feature, an expectation minimization mechanism is added. It inhibits the neuron W_k if it is possible to predict its selection by the WTA based only on $S^p_i, i \neq k$, thus preventing two networks from carrying the same information.

This adaptive WTA achieves good performance in the selection of the best predictor even if the mean value of the similarity measurement varies a lot during an experiment (for instance if the system is presented with some very different kind of input stimuli). By construction, this WTA architecture always outputs a result which can be filtered by looking at the global input activity if necessary.

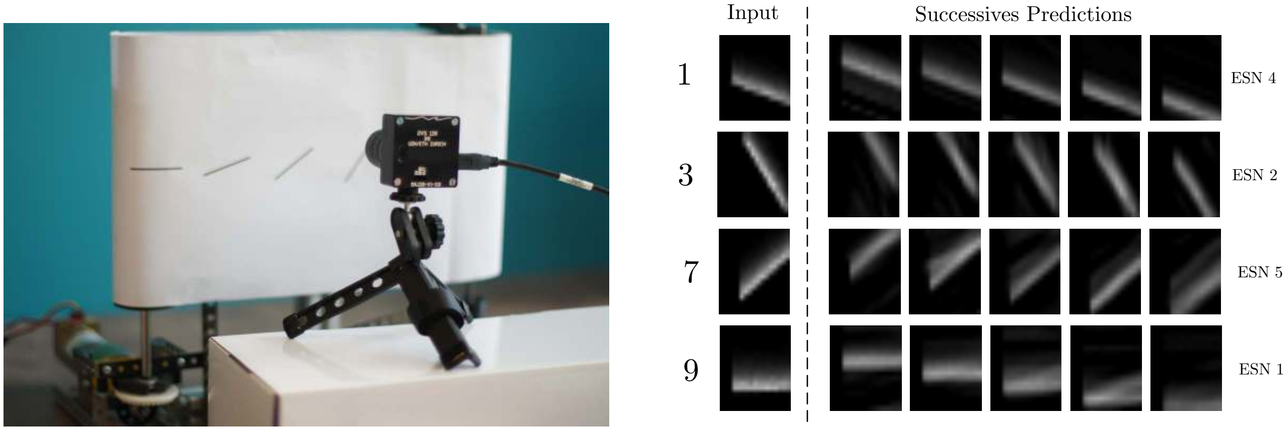
E. Expectation minimization

The layer (2) of Fig. 2 contains the WTA network and an expectation minimization mechanism, which is detailed in Fig. 3. This mechanism ensures that each ESN is specializing in predicting independant features of the input signal. Thus, it implements a criterion as described in [10] to evaluate the relevance of the prediction of each ESN.

An ESN is said to be relevant if its prediction is not redundant with the other ESNs' predictions. For each ESN k, we implement the estimator $\hat{W}_k$ of the WTA output which is only fed by the other ESNs' similarity measures. If the estimator and the WTA outputs are both giving k i.e. $\hat{W}_k$ satifies at t_n:

$$\left\{ \begin{array}{l} |\hat{W}_k(t_n) - W(t_n)| < \epsilon_w \\ W(t_n) = k \end{array} \right. , \qquad (13)$$

where ϵ_w is a threshold determined experimentally, then the k^{th} ESN is not learning a new feature. In this case, the corresponding output of the WTA is inhibited. We use



(a)

(b)

Fig. 4. (a) Experimental setup with a DVS looking at patterns moving on a treadmill. (b) Dynamics learned by some of the ESNs, left column shows a selection from the different input patterns, the others show different prediction obtained from the ESN which specialized in the given pattern. The 5 presented predicted patterns are spaced by 0.01 seconds.

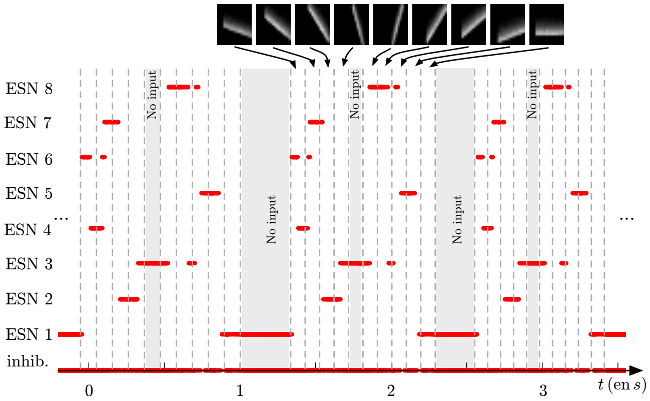


Fig. 5. Output of the WTA network when presented with a repetitive serie of each of the patterns.

ESNs to implement the estimators for the same reason we mentioned in the introduction.

IV. Results

Experimental results are carried out with the DVS and the setup is shown in Fig. 4(a). A treadmill is presenting nine printed bars with different orientations to the camera, at different speeds.

The first experiment is setup with less individual ESNs (8) than features to detect (9 orientations), then we increase the number of ESNs to show how this parameter affects the learning mechanism. Each ESN is made of 15 hidden neurons interconnected randomly and the spectral radius as defined in [4] is set experimentaly to 0.7. To validate the feature extraction process, only one RF is used and composed of 17×17 groups of 5×5 DVS's pixels.

At a given time each ESN outputs a different prediction of the input signal. The WTA succeeds in selecting the correct network corresponding to the best predictor for the current input. The best predictions for 4 of the 8 used ESNs are given in Fig. 4(b). As expected, results show that every network has specialized in the prediction of the temporal evolution of a specific oriented pattern. To be closer to a real world case very rich in term of independent features, we chose to use less ESNs than input features. We can then observe that, because the system does not have enough outputs, some ESNs can learn to code for more than one feature so that the output of the system can represent all possible inputs. It is the role of another upper-level system to process the output of the ESNs and to detect these cases.

This behavior can also be observed in Fig. 5 which summarizes the extraction process on the full length of the input signal. The figure emphasizes the output of the WTA network for three presentations of the stimulus. We can see that each ESN is correctly responding to a particular orientation of the bars. Moreover, the process is repeatable over the three presentations with a difference in the temporal span of the responses. This is due to the increase of the translation speed of the bars during the recording to show that the stimuli velocities have little effect on the network performances.

Fig. 6 plots the prediction error of each ESN during several presentations of the stimulus. Large errors are produced by the ESNs that do not "suit" the stimulus. The ESN giving the smallest prediction error is selected as the best predictor. This is shown in the figure by the red underlining. Note that the time periods where all the prediction errors are close to zero correspond to period without input stimulus.

Fig. 7 emphasizes the repeatability of the network output. This figure presents spikes rasters from the eight ESNs during several presentations of the input stimulus. Vertical dimension shows ten repeated stimulus trials (we limited the representation to ten trials to improve readability). We can see that the response is the same for the ten presentations. We can also notice that for some orientations, an ESN starts to respond but finally another one is the best predictor in the end. For instance, at $t = 0.5s$, ESN 6 starts to respond to the stimulus but the best predictor will be ESN 4. We can also note that ESN 1 is still responding when there is no input.

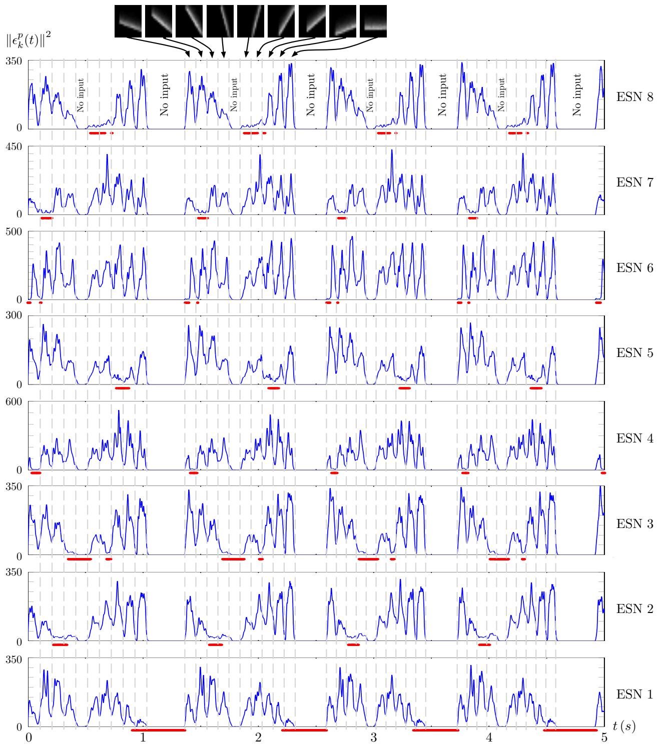
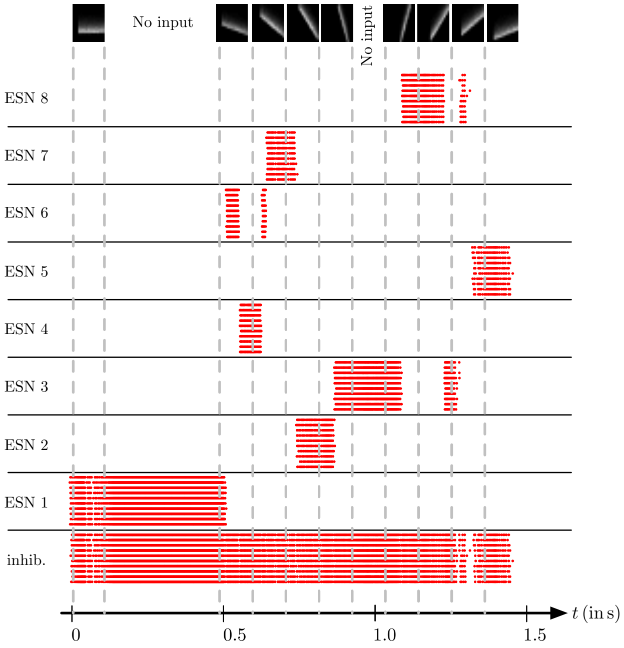
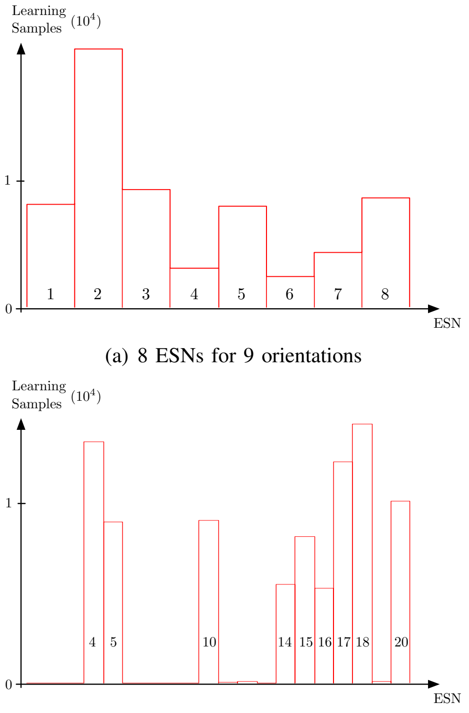


Fig. 6. Prediction error of the different ESNs during several presentations of the input stimulus. The plot shows the evolution of the prediction error for each of the eight ESNs used during the experiment. Under each plot is represented the output of the associated WTA neuron showing when a given predictor is selected.



(a)

Fig. 7. Examples of spikes rasters from the eight ESNs during the presentation of the serie of nine lines. Each dot represents the time of a spike; vertical dimension shows 10 repeated stimulus trials.



(a) 8 ESNs for 9 orientations

(b) 20 ESNs for 9 orientations

Fig. 8. Number of training samples per reservoir for different numbers of used ESNs, showing that when the pool of ESNs is bigger that the number of features present in the input stimulus, only the necessary portion of the pool is used to learn these features and that the rest of the ESNs are left untouched for future features.

This is caused by the fact that ESN 1 is selected by the WTA network which keeps selecting it in absence of input since all predictors' configuration are unchanged.

One fundamental question raised by this architecture is of the number of networks necessary to represent all of the elementary features. Learning statistics are presented in Fig. 8. We can see the number of sampling during which each network has learned some input signal in two cases. First in a setup composed of 8 ESNs for 9 orientations (Fig. 8(a)), we can see that each one of the 8 ESNs was active for a similar number of samples. This underlines the fact that no particular ESN takes over the learning process. This is coherent with the output of the system where one ESN codes for one or several orientations and so there is not enough networks to code all of the elementary features. Fig. 8(b), we still use 9 orientations but now with 20 ESNs. What we can observe is that only 9 of the ESNs were activated during the learning process, each one focusing on a particular orientation.

V. Conclusions

We presented in this paper an event-based architecture to learn and extract features from asynchronous event-based visual streams. The architecture's performances have been tested with different visual stimuli that are presented to the camera at speeds ranging from slow to extremely high where conventional imaging approaches would fail to run in real time. Results show that the method allows to extract features in a reliable and repeatable manner. This is an important property essential to ensure a stable vision-based navigation.

These results open new perspectives to the development of a new kind of bioinspired visual algorithms more robust to environments' conditions but also efficient in energy and computation resources.

REFERENCES

[1] B. Roska and F. Werblin, "Rapid global shifts in natural scenes block spiking in specific ganglion cell types." *Nature Neurosci*, vol. 6, pp. 600–608, 2003.

[2] C. Mead and M. Mahowald, "A silicon model of early visual processing," *Neural Networks*, vol. 1, no. 1, pp. 91–97, 1988.

[3] T. Delbruck, B. Linares-Barranco, E. Culurciello, and C. Posch, "Activity-driven, event-based vision sensors," *ISCAS*, pp. 2426–2429, 2010.

[4] H. Jaeger, "Tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the "echo state netwrok" approach." Tech. Rep., 2002.

[5] B. Schrauwen, D. Verstraeten, and J. V. Campenhout, "An overview of reservoir computing : theory, applications and implementations," in *Proceedings of the 15th European Sympsosium on Artificial Neural Networks*, 2007, pp. 471–482.

[6] S.-C. Liu and M. Oster, "Feature competition in a spike-based winner-take-all vlsi network," in *ISCAS*. IEEE, 2006.

[7] M. Oster, R. J. Douglas, and S.-C. Liu, "Computation with spikes in a winner-take-all network," *Neural Computation*, vol. 21, no. 9, pp. 2437–2465, 2009.

[8] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128X128 120dB 15us latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[9] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004. [Online]. Available: http://www.sciencemag.org/content/304/5667/78.abstract

[10] H. B. Barlow, "Unsupervised learning," *Neural Computation*, vol. 1, pp. 295–311, 1989.