


Spiking Hough for Shape Recognition

Pablo Negri^{1,2} , Teresa Serrano-Gotarredona³,
and Bernabe Linares-Barranco³

¹ CONICET, Dorrego 2290, Buenos Aires, Argentina

² Universidad Argentina de la Empresa (UADE), Lima 717, Buenos Aires, Argentina
pnegri@uade.edu.ar

³ CSIC, Instituto de Microelectronica Sevilla (IMSE-CNM), 41012 Sevilla, Spain

Abstract. The paper implements a spiking neural model methodology inspired on the Hough Transform. On-line event-driven spikes from Dynamic Vision Sensors are evaluated to characterize and recognize the shape of Poker signs. The multi-class system, referred as Spiking Hough, shows the good performance on the public POKER-DVS dataset.

1 Introduction

Histogram based features are a useful tool employed in Computer Vision to describe the content of an image. They are successfully used for object detection associated with different kinds of classifiers. There exists several versions, such as: HOG [4], SIFT [8], HO2L [9], etc.

Recently, a histogram based feature were proposed to characterize the shape of objects captured by an Event-Driven Dynamic Vision Sensor (DVS) [10]. This cameras are inspired on the neuromorphic behavior of the human visual system [7]. They consist of an artificial retina where each pixel captures a light change and generates a spike event. This event is defined as $\mathbf{e} = ((x, y), t, pol)$, (x, y) being the coordinates of the pixel on the grid, t the event time stamp, and pol the polarity. Polarity is a binary ON/OFF output. ON polarity informs an illumination increase, and OFF polarity is obtained when illumination decreases. An event flow composed of N consecutive events is defined as: $\mathbf{w}_N = \{\mathbf{e}_1, \dots, \mathbf{e}_N\}$. This kind of vision sensors is considered as “frameless” providing asynchronous high temporal resolution data.

Others histograms representations using DVS events flows were proposed in the literature. Clady *et al.* [3] proposed a hand-gesture recognition framework using a histogram representation of flow motion vectors. In [5] they employed a hierarchical model architecture (HOTS) consisting of several consecutive layers of increasing detail and including a histogram representation of time-surface activations for each object class. The architecture is based on a deep neural network, similar to the Convolutional Spiking Neural Network in [11, 18] for object recognition.

The proposed system is denominated Spiking Hough, because it is inspired on the Hough Transform and uses a spiking neural network approach. In [6, 12]

the Hough transform is applied on the DVS outputs for straight edges and lane detection. On the other hand, this paper proposes a methodology which captures the spatial distribution of the events generated in the retina, and organize this information in histogram features. The histograms feed a multi-class classifier to recognize the shape of the objects.

The paper is organized as follow. Next section details the Spiking Hough approach and the classification system. Section 3 presents and discusses the results. The conclusions in Sect. 4 resume the work and propose some perspectives.

2 Spiking Hough Multi-class Classification System

The systems aims to classify the events flow into one of the four Poker signs. The pipeline is presented in Fig. 1. Incoming events spike the neurons distributed on the retina grid. The Spiking Hough receives the firing neurons, capturing the configuration of the object’s shape and builds the cell histograms. The framework counts the number of incoming events, and when this number reaches N , the histograms features are evaluated with a multi-class Support Vector Machine (SVM) [15] classifier. After the classification, the histograms are reset for the next windows event.

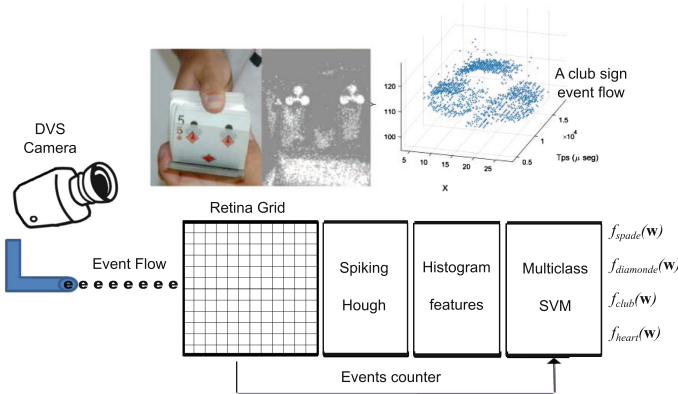


Fig. 1. Poker-DVS dataset extraction. Left image shows a RGB capture (image from [11]), and right image shows the DVS retina captured events of a *club* sign.

2.1 Spiking Neural Network Model

Figure 2 shows a neuron model [1]. In this model, dendrites are the inputs terminals to the nucleus, and the axons their outputs. The neural interfaces between dendrites and axons are denominated synapses.

The dynamic of the neuron model is controlled by input spikes, an electrical signal with short duration, arriving through the dendrites. The potential of these

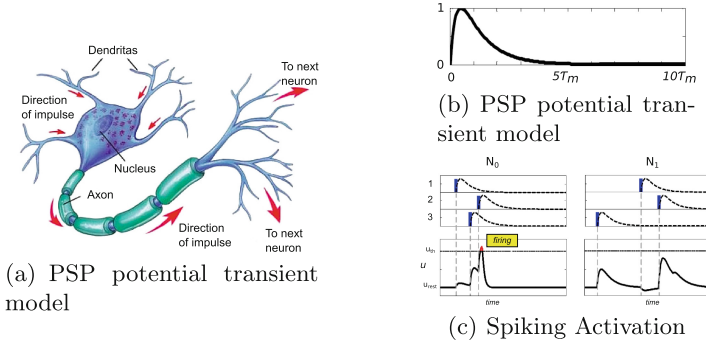


Fig. 2. (a) Neuron model [1], (b) PSP transient model and (c) Spiking neuron model (from [17]) (Color figure online)

incoming spikes are accumulated in the neuron membrane. When the potential reaches a threshold, the neuron fires a signal, an output spike, to the next neurons via the axons. Synapses control the amplitude of the spike traveling to the following neuron.

The potential of the membrane receiving an excitation from a spike has a transient behavior commonly referred as Post-Synaptic Potential (PSP). PSP can affect the membrane's potential by: Excitatory PSP (EPSP) which increments the potential, or Inhibitory PSP (IPSP) which decrements the potential. Figure 2(b) shows the PSP potential computed using the Integrate-and-Fire approach, with equation $u(t - t_i) = V_0(e^{-\frac{t-t_i}{\tau_m}} - e^{-\frac{t-t_i}{\tau_s}})$, where τ_m and τ_s denote decay time constants of the membrane integration, and V_0 normalizes the potential so that the maximum value is 1.

The spiking neural model is shown in Fig. 2(c). Two neurons, N_0 and N_1 , with three afferent synapses inputs. The spikes on the input synapses are shown as blue signals. After the spike, the PSP model shows the individual afferent transient potential added to the membrane potential $u_j(t)$. For N_0 all PSP signals correspond to excitatory PSPs, with different synapses weights. In the case of N_1 synapses 1 corresponds to a IPSP which provides a negative potential to $u_1(t)$. In the example, after the three spikes of their afferent synapses, the potential of N_0 reaches threshold u_{th} and generating an output spike. Then, the potential $u_1(t)$ is reset to the u_{rest} value. Neuron N_1 did not fire, because their potential never crosses the threshold in the temporal window.

Spiking neural models are well adapted to be employed with the DVS data flow. The flow of individual events feeds the neurons which are trained to trigger when a special configuration is detected. The spiking model is adapted to a Hough analysis in the next section.

2.2 Hough Transform Analysis

The Hough Transform projects data from the image xy -plane to a new $\theta\rho$ -plane. In this space, the equation of a single line passing through the point (x_0, y_0) ,

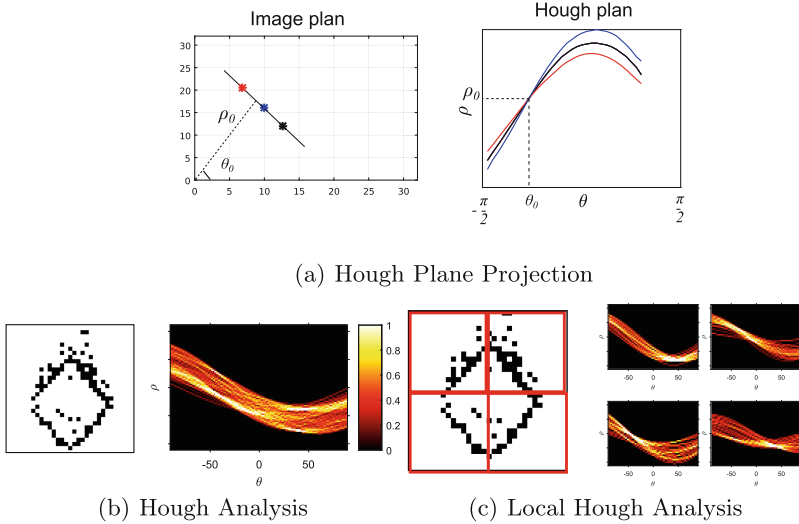


Fig. 3. Original Hough Transform straight lines projection. Hough transform on a $w_{N=200}$ windows event of a diamond sign.

Fig. 3(a), can be written as: $x \sin \theta_0 + y \cos \theta_0 = \rho_o$, which corresponds to a point in the $\theta\rho - plane$. Thus, the infinite lines which pass through (x_0, y_0) generate a curve in the Hough plane, Fig. 3(a). The Hough Transform identifies when several points lying on the same line. The curves on the Hough plane generated by the points intersects at (θ_0, ρ_0) in the $\theta\rho - plane$, which are the parameters of the line. Computationally, the Hough Transform converts the $\theta\rho - plane$ into a so-called *accumulator cell*. A cell (θ_i, ρ_i) receiving a high number of votes shows that several points in the $xy-plane$ lay on the line with this parameters.

Because Hough Transform is based on a votes algorithm it is robust against noise. In Fig. 3(b), voting cells in the Hough space having maximal votes are placed at $\theta = -\pi/4$ and $\theta = \pi/4$ as expected for the diamond shape. The two maximums for each orientation corresponds to the two different ρ values of the two edges with the same orientation. In Fig. 3(c) the 32×32 pixels retina was split in four cells. This allows a local analysis of the edges, which is in fact a more discriminant way to describe the shape. The four Hough accumulators, one for each cell, are dominated for a maximum value placed at the orientation associated with the diamond shape edge.

2.3 Spiking Hough for Feature Extraction

The Spiking Hough defines a set of 12 neurons inspired on the Hough parameter space. Such space, referred as Spiking Hough Space, is discretized by four orientations and three biases, as shown Fig. 4(a). The neurons fire when the sequential events spikes at all the gray positions, and then reset their potential membrane.

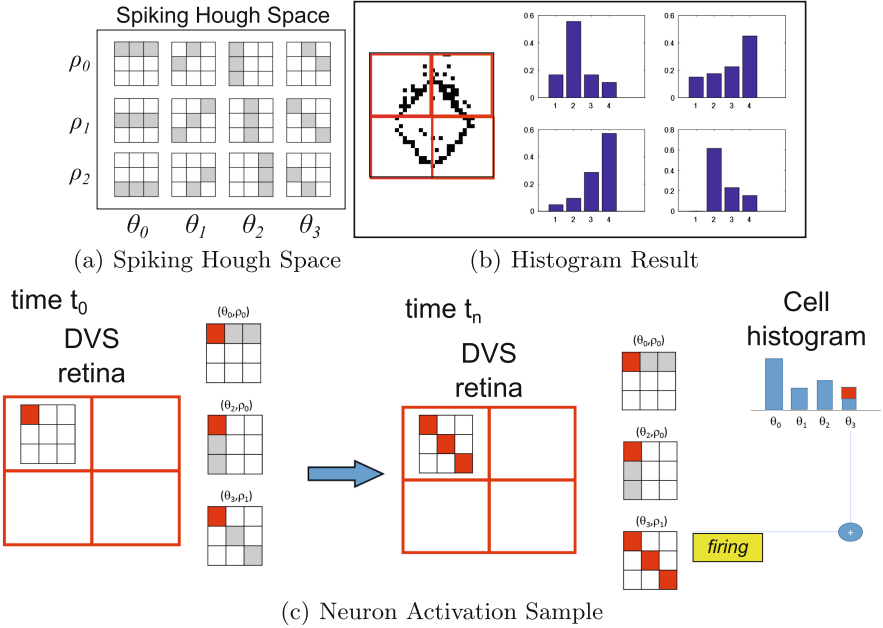


Fig. 4. (a) Spiking Hough Space containing 12 neurons specialized on each orientation, (b) Example of the activation of a neuron and the building of the cell histogram, (c) Histogram feature result for the diamond shape using $N = 200$ events. (Color figure online)

The poker signs are contained in retina of 32×32 pixels size. Similar to Sect. 2.2, the Poker retina is subdivided into four non-overlapped cells. Each cell has 16×16 pixels size. Inside each cell, overlapped patches of 3×3 elements receive the events spikes. There are 36 overlapped patches inside each cell to cover all their pixels. To be robust against the stochastic nature of the DVS, each element in the 3×3 patch corresponds to 2×2 pixels of the retina.

Spiking Hough identifies the predominant edges configuration by firing a corresponding neurons on its space. Figure 4(c) illustrates the firing of a neuron. At time t_0 there arrives an event e_0 at the cell lying on the 3×3 patch corner, painted in red. This event spikes the neurons with this position activated (in gray), increasing their potential. A sequence of events produce spikes on others positions of the patch. Later, at time t_n , an event e_n fires the neuron with (θ_3, ρ_1) . Then, the firing neuron triggers and increments the 3rd. bin of the cell histogram, which correspond to the orientation of the detected edge.

2.4 Supervised Classification

The multi-class SVM classifier framework was trained using the LIBSVM library [2] and the best parameters for the linear and the RBF kernels were estimated

using a 5 cross-fold validation approach. The framework was composed of the four SVM classifiers, trained using the one-against-one approach. LIBSVM uses [16] to obtain a single probability score for each class k : $f_k^{svm}(\mathbf{w}_i)$, with $k = 1, 2, 3, 4$.

$$P_k(\mathbf{w}_N^i) = \alpha f_k(\mathbf{w}_N^i) + (1 - \alpha) f_k(\mathbf{w}_N^{i-1}) \quad (1)$$

$$k^* = \operatorname{argmax}_{k=1,2,3,4} P_k(\mathbf{w}_N^i) \quad (2)$$

For an input window \mathbf{w}_N^i , the probability to belong to sign k is computed with Eq. 1, where α is a memory factor. Thus $P_k(\mathbf{w}_N^i)$ uses the current and previous event window classification functions $f_k(\mathbf{w}_N^i)$ and $f_k(\mathbf{w}_N^{i-1})$ to smooth the response and become robust to noisy windows. Sample \mathbf{w}_N^i is classified as in class k^* which $P_{k^*}(\mathbf{w}_N^i)$ produces the largest probability output on Eq. 1.

3 Experiments and Results

The Spiking Hough classification system was conducted on the 2015 Poker-DVS dataset [13]. On their website, the authors share a complete recording of the asynchronous events while they were browsing the poker cards, as well as a set of 131 individual files of cropped events. Each file has a name indicating the sign to which the flow of events corresponds. A character ‘i’ is added if the card is inverted. There are 30 club signs (13 inverted), 43 diamonds (8 inverted), 23 hearts, and 35 spades (10 inverted).

Given the low number of samples per class, the tests were conducted using the Leave-One-Out approach. This methodology employs all the samples of the set to train the multi-class classifier, except for one sample which is evaluated by the classifier and the result is saved in a confusion matrix. The overall performance is then obtained by computing the accuracy on the diagonal of the matrix.

The event flow of a test sample feeds the Spiking Hough system until the number of events reaches N . This event window is referred \mathbf{w}_N^0 and is then evaluated by the four SVM classifiers using Eqs. 1 and 2, and $\alpha = 0.5$ (which gives the best results on the tests). In this way, the output of the classification accumulates votes for each sign. This procedure is repeated to the next events of the sign, until the end of the flow, and the output of each \mathbf{w}_N^i is evaluated with Eqs. 1 and 2. The test sample is finally classified by the sign that receives the highest number of votes.

Linear and Radial Basis Function (RBF) kernels are used to implement the SVM multi-classification. The results of the linear and non-linear (RBF) kernels are compared on Fig. 5. There were also tested to cell grids. One with 4 non-overlapped cells, as shows Fig. 4(b). The other configuration incorporates 4 more cells overlapping the original ones, to obtain 8 cells in total. The last configuration helps the system to be robust against little movements of the shape, and the fact that the 32×32 Poker retina is not necessarily centered all the time. The length of the event window N was also evaluated, from a minimum of 100 events to a maximum of 500. The Figure also shows the associated time delays representing the different lengths of event windows. It was calculated as the average value of all the \mathbf{w} in the dataset for a specified N .

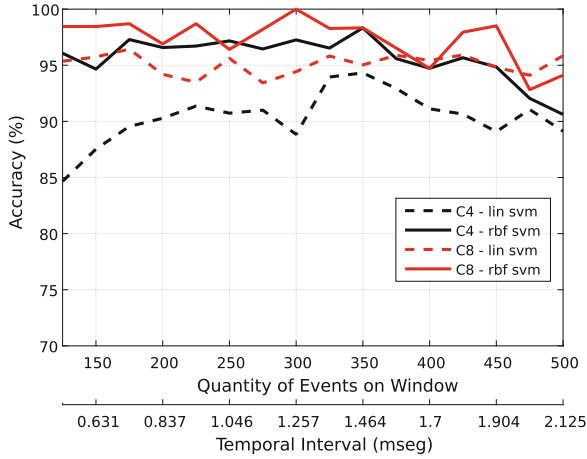


Fig. 5. Results of different classifiers and features extractions approaches.

The systems shows a good robustness, even if there exist several signs that are inverted. The non-linear RBF kernel on the configuration of 8 cells using $N = 300$ obtains an accuracy of 100%. This result outperforms the best performance published in the state of the art [14]. For this case, all the Poker signs were correctly classified. From the temporal axis, this event windows lengths corresponds on average to 1.256 *mseg*.

4 Conclusions

The Spiking Hough methodology describes *on-line* the shape of the events generated by an object moving in front of the DVS camera. The resulting histograms features show a good discriminating power for the Poker sign recognition.

Further research should be oriented to obtain a complete system which works on-line from the feature generation until the classification. Also, the system must be prepared to moving object and different scales in order to be though as a good choice to be implemented in real applications.

Acknowledgements. This work was funded by PID Nro. P16T01 (UADE, Argentine), EU H2020 grants 644096 “ECOMODE” and 687299 “NEURAM3”, and by Spanish grant from the Ministry of Economy and Competitvity TEC2015-63884-C2-1-P (COGNET) (with support from the European Regional Development Fund).

References

1. Anderson, J.: An Introduction to Neural Networks. MIT Press, Cambridge (1995)
2. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. IST **2**(3) (2011). <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

3. Clady, X., et al.: A motion-based feature for event-based pattern recognition. *Front. Neurosci.* **10**, 594 (2017)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *CVPR*, vol. 1, pp. 886–893 (2005)
5. Lagorce, X., et al.: HOTS: a hierarchy of event-based time-surfaces for pattern recognition. *PAMI* **39**(7), 1346–1359 (2017)
6. Li, X., et al.: Lane detection based on spiking neural network and hough transform. In: *CISP*, pp. 626–630 (2015)
7. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128*128 120dB 15us latency asynchronous temporal contrast vision sensor. *JSSC* **43**(2), 566–576 (2008)
8. Lowe, D.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
9. Negri, P.: Pedestrian detection using multi-objective optimization. In: Pardo, A., Kittler, J. (eds.) *CIARP 2015*. LNCS, vol. 9423, pp. 776–784. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25751-8_93
10. Negri, P.: Extended LBP operator to characterize event-address representation connectivity. In: Beltrán-Castañón, C., Nyström, I., Famili, F. (eds.) *CIARP 2016*. LNCS, vol. 10125, pp. 241–248. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-52277-7_30
11. Pérez-Carrasco, J., et al.: Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing-application to feedforward convnets. *PAMI* **35**(11), 2706–2719 (2013)
12. Seifozzakerini, S., et al.: Event-based hough transform in a spiking neural network for multiple line detection and tracking using a dynamic vision sensor, pp. 94.1–94.12, September 2016
13. Serrano-Gotarredona, T., Linares-Barranco, B.: 2015 poker-DVS dataset (2015). <http://www2.imse-cnm.csic.es/caviar/POKERDVS.html>. Accessed 8 June 2017
14. Stromatias, E., Soto, M., Serrano-Gotarredona, T., Linares-Barranco, B.: An event-driven classifier for spiking neural networks fed with synthetic or dynamic vision sensor data. *Front. Neurosci.* **11**, 350 (2017)
15. Vapnik, V.: *The Nature of Statistical Learning Theory*. Springer, New York (1995). <https://doi.org/10.1007/978-1-4757-3264-1>. ISBN 9780387987804
16. Wu, T.F., Lin, C.J., Weng, R.: Probability estimates for multi-class classification by pairwise coupling. *J. Mach. Learn. Res.* **5**, 975–1005 (2004)
17. Zhao, B., et al.: Event-driven simulation of the tempotron spiking neuron. In: *BioCAS*, pp. 667–670, October 2014
18. Zhao, B., et al.: Feedforward categorization on AER motion events using cortex-like features in a spiking neural network. *Neural Netw. Learn. Syst.* **26**(9), 1963–1978 (2015)