

# Edge Detection in Time Variant Scenarios Based on a Novel Perceptual Method and a Gestalt Spiking Cortical Model

Juan Ramírez-Quintana<sup>(✉)</sup>, Mario Chacon-Murguia,  
and Alma Corral-Saenz

PVR Lab, Chihuahua Institute of Technology, Chihuahua, Mexico  
{jaramirez, mchacon, adcorral}@itchihuahua.edu.mx

**Abstract.** Based on recently neurocomputational models inspired on neural synchronization for perceptual grouping, we propose in this paper the Gestalt Spiking Cortical Model (GSCM) and the Perceptual Grouping segmentation (PGSeg). The GSCM is a network based on the mechanisms of perceptual grouping models designed to detect scene attributes with excitatory and inhibitory inputs. PGSeg is a neuroinspired method designed to detect object edges presented in video sequences that involve time variant scenarios. Experimental results using videos from the perceptual computing and ChaDet2014 databases, show that PGSeg has better performance regarding edge detection and edge coherence through video sequences.

**Keywords:** Spiking neural networks · Edge detection · Background modeling

## 1 Introduction

Object perception is a complex human visual perception ability that involves interpretation of multiple features such as motion, depth, color and contours. Among these features, contour integration is a special case of perceptual grouping generated in the lower layers of the visual cortex that allows psychophysiological measures which contributes to integrate other features such as depth and movement [1]. There are many theories about how neurons develop visual perceptions skills, and some of them postulate that neural groups represent object features through synchronization of their pulse activity [2]. This synchronization plays an important role in perceptual grouping and Gestalt principles [3]. Based on these theories, different models have emerged such as oscillatory neural networks and Spiking Neural Networks (SNN). Among them, Pulse-Coupled Neural Networks (PCNN) have been used on contour integration, edge detection and other similar applications in image processing [4]. Several models based on PCNN have been reported in the literature to deal with these applications [5–7]. However, most of them do not consider the Gestalt principles given by neural synchronization and do not consider the most recently theories about contour perception in visual cortex. Therefore, we propose in this paper the Gestalt Spiking Cortical Model (GSCM), a PCNN simplified model based on the Gestalt rules generated from neural synchronization. Furthermore, we developed the method termed Perceptual Grouping

Segmentation (PGSeg) that applies the GSCM model to perform edge objects detection in a similar way to the neurocomputational models that describes the process in the lower layers of visual cortex for contour integration and perceptual grouping. Edges caused by changes in object color, dynamic background conditions, lines inside objects, reflections and shadows are not considered contours of objects. PGSeg was designed for coherent edge object detection in video sequences that involve time variant scenarios, while classical edge detection methods were designed for still images with time invariant scenes. Furthermore, PGSeg consider background modeling of complex scenarios while other spatio-temporal edge detection methods as the presented in [8], consider only static backgrounds. In this paper, edge object detection refers to detect only contours of objects. This paper is organized as follows. Section 2 describes the GSCM. Section 3 describes the PGSeg method. Section 4 shows the results and Sect. 5 describes the conclusions.

## 2 Gestalt Spiking Cortical Model (GSCM)

Based on the perceptual models presented in [2, 3], and stimulus and inhibitor connections in the internal activity of Perceptual Grouping LISSOM presented in [1], the GSCM was defined as follows:

$$U(x, y, n) = U(x, y, n - 1) \cdot \exp(-\alpha_F) + S(x, y, n)(1 - Y(x, y, n - 1) * W_S) - I(x, y, n)(1 - Y(x, y, n - 1) * W_I) \quad (1)$$

$$Y(x, y, n) = \begin{cases} 1 & U(x, y, n) > E(x, y, n) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$E(x, y, n + 1) = E(x, y, n) \cdot \exp(-\alpha_E) + Y(x, y, n) \quad (3)$$

where  $(x, y)$  is the pixel position in the frame,  $n$  is the iteration index,  $U(x, y, n)$  is the internal activity of a neuron,  $E(x, y, n)$  is the dynamic threshold and  $Y(x, y, n)$  is the neuron response.  $\alpha_F$  and  $\alpha_E$  are the exponential decay factors of  $U(x, y, n)$  and  $E(x, y, n)$  respectively.  $U(x, y, n)$  has two inputs: stimulus input  $S(x, y, t)$  and inhibitor input  $I(x, y, n)$ .  $W_S$  is the synaptic weights of the  $S(x, y, n)$  and  $W_I$  is the synaptic weights of  $I(x, y, n)$ . The weights of GSCM are Gaussians defined by:

$$W(\sigma_v) = w(x, y, \sigma_v) = \exp\left(-\frac{(x - x_\omega)^2 + (y - y_\omega)^2}{\sigma_v^2}\right) - \delta(x_\omega, y_\omega), \quad v = \{S, I\} \quad (4)$$

$(x_\omega, y_\omega)$  is the center of weights,  $\sigma_v$  is the neighborhood radius, which depends of the scenario conditions. The Gaussian behavior is because of [2, 3], that indicates that Gestalt rules such as similarity can be implemented with Gaussians connections between oscillating neurons.

### 3 Perceptual Grouping Segmentation Method

PGSeg is a hierarchical method and is illustrated in Fig. 1. The first layer is the input that corresponds to a frame of a video sequence. The second layer is a module inspired on lateral geniculate nucleus (LGN) of the visual cortex that performs an edge soft detection. The third layer is based on the behavior of Orientation Receptive Fields (ORF) of the primary visual cortex. The aim of the ORF layer is to generate an orientation map and to improve the edge soft detection, which will be the input patterns for next layer. The following layer called perceptual grouping, finds the object edges using two GSCM networks: the first GSCM is used to model the background of the video sequence, and the second one detects the lines that are going to be the input to the edge detection layer. PGSeg includes background modeling of complex scenarios, therefore it may seem more complicated than classic edge detection methods. In the next subsections all layers will be explained.

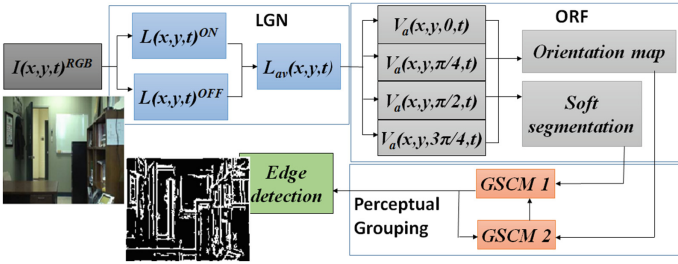


Fig. 1. PGSeg method.

The experiments were performed using different videos from the perceptual computing ([http://perception.i2r.a-star.edu.sg/bk\\_model/bk\\_index.html](http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html)) and ChaDet2014 ([changedetection.net](http://changedetection.net)) databases, which are popular in literature. The selected videos have time-variant scenarios with different conditions, which are described in [9].

#### 3.1 Input Layer

The input layer acquires a frame  $I(x, y, t)^{RGB}$  of a video sequence, where  $t$  is the frame index. PGSeg does not require color information, therefore, the input layer extract the Value component ( $V$ ) of HSV color space computed from the RGB information.

#### 3.2 Lateral Geniculate Nucleus Layer

The LGN layer is inspired in the Receptive Fields (RF) located in the Retinal Ganglion Cells to Lateral Geniculate Nucleus on the visual cortex. Those RFs are modeled as simple cells whose response is given by:

$$RF(x, y)^c = \sum_a \sum_b I(x - a, y - b, t)^V G(a, b), \quad c = \{ON, OFF\} \quad (5)$$

where  $G(x, y)$  is the response of the RFs, given by difference of gaussians (DoG) defined in [10]. In order to simplify the method, PGSeg uses for next layers arithmetic average of  $L(x, y, t)^{ON}$  and  $L(x, y, t)^{OFF}$ ,  $L_{av}(x, y, t)$ . Figure 2 shows the response of  $L_{av}(x, y, t)$ , which is a feature map that describes edges generated by the objects of the scenario.



**Fig. 2.**  $L_{av}(x, y, t)$  response. (a) Streetlight video,  $t = 100$ . (b)  $L_{av}(x, y, t)$ .

### 3.3 Orientation Receptive Fields Layer

After the LGN layer process, the visual information derived from RFs is projected to the primary visual cortex (V1). V1 has a cortical layer that consists on a set of receptive fields selective to orientation features (ORFs) [11]. Based on these ORFs, we design for PGSeg a layer with a set of orientation selective filters which were inspired in the model presented in [10], and defined by:

$$ORF(x, y, \theta) = \exp\left(-\frac{[(x - x_c) \cos \theta - (y - y_c) \sin \theta]^2}{\sigma_d^2} - \frac{[(x - x_c) \cos \theta + (y - y_c) \sin \theta]^2}{\sigma_f^2}\right) \quad (6)$$

where  $ORF(x, y, \theta)$  is selective to lines with orientations similar to  $\theta$ ,  $(x_c, y_c)$  is the center of the filter,  $\sigma_d$  and  $\sigma_f$  define the size of width and length of the filter. The ORF are modeled as simple cells [11] and defined with:

$$V_a(x, y, \theta, t) = L_{av}(x, y, t) * RFO(x, y, \theta) \quad (7)$$

In PGSeg, the orientation filters are selected with orientation  $\theta = \{0, \pi/4, \pi/2, 3\pi/4\}$ ,  $\sigma_d = 3$  and  $\sigma_f = 1$ . The values of  $\theta$  were selected to simplify the calculus (as  $S_I$  units of HMAX model in [12]),  $\sigma_d$  and  $\sigma_f$  were defined by experimentation. In models such as HMAX [12], the next layer of ORF is a set of nonlinear cells that select the highest magnitude. In the same way, PGSeg finds the ORF with the highest magnitude to have a better response of the edges, as follows:

$$V_{rfo}(x, y, t) = \max(V_a(x, y, \theta, t)) \quad (8)$$

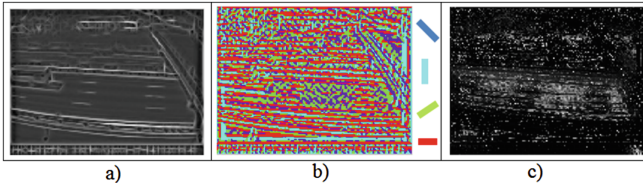
In addition, PGSeg defines an orientation map, as in the LISSOM models [10], which are generated by obtaining the orientation with the greatest magnitude as follows:

$$I_{\theta}(x, y, t) = \max\left(\left(\frac{4}{\pi}\right)\theta(x, y, t) + 1\right) \quad (9)$$

$\theta(x, y, t)$  is the orientation value in each pixel. Figure 3 shows the response of  $V_{rfb}(x, y, t)$  and  $I_{\theta}(x, y, t)$  (orientations are coded with colors). For pixels that belong to edges (edges could be noise or edge objects),  $I_{\theta}(x, y, t)$  has the same value through the time. For the rest of the pixels,  $I_{\theta}(x, y, t)$  have random values through the time. Therefore, it is possible to find the parts of the scenario where there are edges if we analyze differences of  $I_{\theta}(x, y, t)$  and  $I_{\theta}(x, y, t - 1)$ . PGSeg analyzes those differences in order to find edges of the scenario that can be edges of objects. Then, next accumulator is computed:

$$AI_{\theta}(x, y, t + 1) = AI_{\theta}(x, y, t) + |I_{\theta}(x, y, t) - I_{\theta}(x, y, t - 1)| \quad (10)$$

On initial conditions, pixels in  $AI_{\theta}(x, y, t)$  are zero. After processing,  $AI_{\theta}(x, y, t)$  has values close to zero in pixels that belongs to edges. This information is used by PGSeg to find possible edges. Figure 3(c) shows in black parts of the scenario where there are different edges.  $V_{rfb}(x, y, t)$  and  $AI_{\theta}(x, y, t)$  are the input patterns for the next layer, which will be discussed later.



**Fig. 3.** *StreetLight* video response at the frame  $t = 100$ . (a)  $V_{rfb}(x, y, t)$  (b)  $I_{\theta}(x, y, t)$ . (c)  $AI_{\theta}(x, y, t)$ .

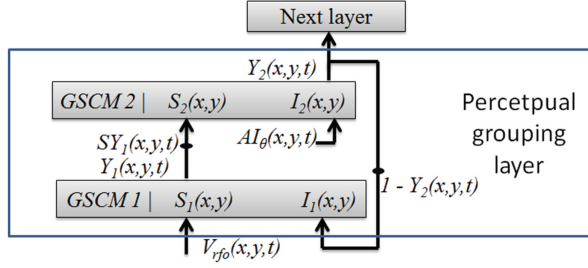
### 3.4 Perceptual Grouping Layer

The perceptual grouping layer has two GSCM as Fig. 4 shows. The first GSCM (GSCM1) is used for background modeling, and the second one (GSCM2) is used to classify the scenario in two classes: object-edges and no-edges. GSCM1 iterates once on each frame ( $n = t$ ), but in the case of the GSCM2, iteration index is restarted each eight frames. The weights  $W_S$  and  $W_I$  of PGSeg were defined based on LEGION [12] and PG LISSOM model [1], in which, there are local excitatory and global inhibitory connections. Then, for PGSeg we must have  $\sigma_I > \sigma_S$ , and according to the experiments,  $\sigma_I = 2$  and  $\sigma_S = 0.5$ .

The input stimulus  $S_I(x, y)$  of the GSCM1 is  $V_{rfb}(x, y, t)$  and for the inhibitory input  $I_I(x, y)$  is  $1 - Y_2(x, y, t)$ , where  $Y_2(x, y, t)$  is the output of the GSCM2.  $V_{rfb}(x, y, t)$  contains a soft detection of lines and edges, and  $Y_2(x, y, t)$  is used to reduce noise, which will be discussed later. The output of the GSCM1,  $Y_1(x, y, t)$ , is a set of pulses which are summed over time to generate the background modeling of  $V_{rfb}(x, y, t)$ . In consequence, the background model is obtained by

$$SY_1(x, y, t) = E_1(t)SY_1(x, y, t - 1) + Y_1(x, y, t) \quad (11)$$

Initially,  $SY_1(x, y, t)$  is zero,  $E_1(t)$  in the range of  $0 < E_1(t) < 1$  is an entropy difference measure that depends on changes of the scenario composition and conditions.



**Fig. 4.** Architecture of the GSCM module.

If there are changes in the scenario composition,  $E_I(t)$  decreases, causing a faster background update since  $SY_I(x, y, t)$  has less influence in the background modeling than  $AY_I(x, y, t)$ . If there are no changes in the scenario composition,  $E_I(t) \approx I$ , then,  $SY_I(x, y, t)$  has more influence than  $Y_I(x, y, t)$  to model the background.  $E_I(t)$  is given by:

$$E_I(t) = \frac{1}{2}(|e(t) - e(t-1)| - (|e(t) - e(t-1)| - 1) + 1) \quad (12)$$

$e(t)$  is the entropy of each frame, which, according to [13] is related to the composition change in the scenario. Figure 5 shows the results of background modeling.



**Fig. 5.** Background modeling. (a) StreetLight frame  $t = 100$ . (b)  $SY_1(x, y, t)$ .

In GSCM2, the input to  $S_2(x, y)$  is  $SY_I(x, y, t)$  normalized in range  $[0,1]$ , and the input to  $I_2(x, y)$  is  $I - AI_\theta(x, y, t)$ .  $SY_I(x, y, t)$  is used to generate edges in  $Y_2(x, y, t)$ , while  $AI_\theta(x, y, t)$  inhibits neurons in  $Y_2(x, y, t)$  connected to regions that do not contain information of the edge objects candidates. The output of  $Y_2(x, y, t)$  respect time is the following. In first iteration all neurons are activated in  $Y_2(x, y, t)$ . In second iteration few neurons are activated or in some cases none is activated. Neurons connected to pixels associated with dark-light changes are activated in the third iteration, and neurons connected to pixels associated with light-dark changes are activated in the fourth iteration. In the next three iterations, neurons associated to pixels with noise are activated. However, neurons in  $Y_2(x, y, t)$  connected to regions that do not have edges, have no response after first iteration because of the inhibition of  $AI_\theta(x, y, t)$ . The next iterations have same behavior but with more noise. Therefore, every time that  $t$  is a multiple of eight,  $Y_2(x, y, n)$  is restarted.  $Y_2(x, y, n)$  is used as a feedback factor in the GSCM1 as shown in Fig. 4, because it helps to improve the response in  $Y_I(x, y, t)$  since  $Y_2(x, y, t)$  can help to inhibit the response in neurons connected to pixels with noise in

$V_{rfo}(x, y, t)$  related to edges caused by changes in object color, dynamic background conditions, lines inside objects, reflections and shadows.

### 3.5 Edge Detection Layer

The information obtained by the pulses of iterations of light-dark and dark-light edges is used for edge objects detection. Therefore, the background contours are obtained by

$$Y_{acum}(x, y, t) = \sum_{p=3,4} Y_2(x, y, p) \quad (13)$$

$$I_s(x, y, t) = \frac{1}{2}(|Y_{acum}(x, y, t)| - |(Y_{acum}(x, y, t)) - 1| + 1) \quad (14)$$

where  $p$  is the module of  $t/8$ ;  $p = \{3, 4\}$  are the iterations with contours information (edge objects). The iterations  $p = \{1, 2, 5, 6, 7\}$  are related to the class no-edges. Figure 6 shows the result of PGSeg with an outdoor scenario.

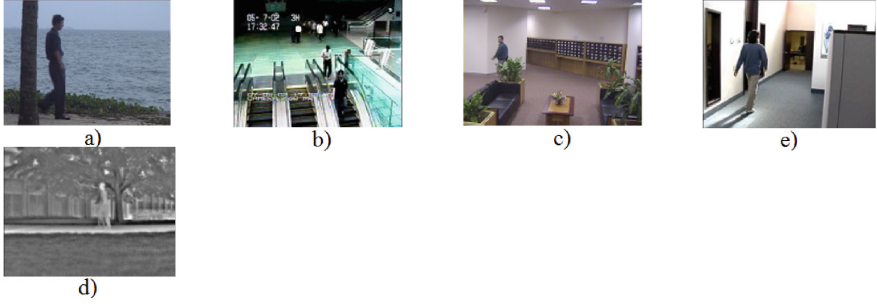


**Fig. 6.** Edge detection with PGSeg. (a) *StreetLight*, frame  $t = 100$ . (b)  $I_s(x, y, t)$ .

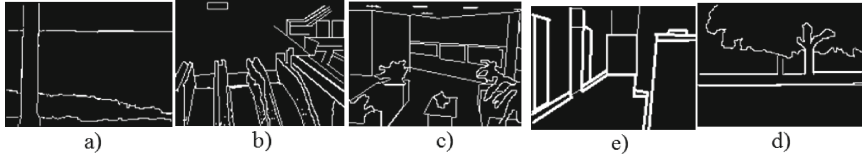
## 4 Results of PGSeg

The metric F measure (F1) [14] was used to compare the performance of PGSeg with the Canny, LoG, Roberts, Prewitt and Sobel methods which sometimes had been used for comparison purposes in literature [15–18]. This comparison consists on obtaining F1 between the ground truth and each  $I_s(x, y, t)$  resulting from the PGseg processing. Then, the average of each F1 of  $I_s(x, y, t)$  is obtained ( $\mu F1$ ) for each video sequence. The videos used to measure the performance, were selected based on the situations that generate time-variant scenarios, and those videos are: Watersurface (WS), Subway Station (SS), Lobby (LB), Cubicle (CU) and Park (PK). The WS, SS and LB videos were obtained from the data base of Perceptual Computing and the CU and PK videos were obtained from ChaDet2014. The ground truths are images of edges that represent the contours of background objects. The edges caused by changes in object color, dynamic background, lines inside objects, reflections and shadows are considered false edges in this work. Figure 7 shows a frame of each video sequence and Fig. 8 shows the ground truths.

As Fig. 8 shows, all the videos have people crossing the scenario, as dynamic objects. Moreover, each video has different situations: in the case of the WS video, the sea generates a dynamic background on the scenario; the SS video has a dynamic background caused by the electric stairs and the light reflections in the floor could cause false edges;



**Fig. 7.** Video sequence for validation. (a) ‘WS’ video,  $t = 500$ . (b) ‘SS’ video,  $t = 500$ . (c) ‘LB’ video,  $t = 370$ . (d) ‘CU’ video,  $t = 5000$ . (e) ‘PK’ video,  $t = 500$ .



**Fig. 8.** Ground truths. (a) ‘WS’ video. (b) ‘SS’ video. (c) ‘LB’ video. (d) ‘CU’ video. (e) ‘PK’ video.

the LB video has sudden illumination changes; the CU video has several shadows that can cause false edges. Finally, the PK video was recorded with a thermal camera with camouflage issues. Table 1 shows the results of  $\mu FI$  for each video sequence, where PGseg has the better results. The parameters of Canny, LoG, Roberts, Prewitt and Sobel were selected based on the best  $\mu FI$  results for each method in each video.

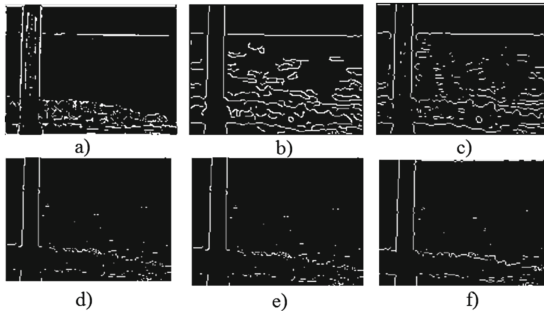
**Table 1.** Results of  $\mu FI$ .

Video/method	PGSeg	Canny	LoG	Roberts	Prewitt	Sobel
WS	0.5798	0.23093	0.28701	0.28149	0.33062	0.3309
LB	0.2677	0.2551	0.1996	0.1795	0.1825	0.1825
SS	0.2761	0.2525	0.195	0.1331	0.1632	0.1626
CU	0.5801	0.4892	0.3806	0.2717	0.2718	0.2755
PK	0.3095	0.2382	0.2687	0.2573	0.2072	0.2081

Figure 9 shows the results of each method with the WS video. PGSeg has adequate results, but with a few noise. However, Canny and LoG generates noise due dynamic background, also, the edge detection is different between one frame and another, although the scenario does not have changes its composition. The Roberts, Prewitt and Sobel methods fail detecting the edges and generate noise. PGSeg generates better results because  $Y_2(x, y, t)$  inhibit the noise in the results in the background modeling, and  $AI_\theta(x, y, t)$  inhibits the neurons in  $Y_2(x, y, t)$  that are connected to object that generate false edges in the dynamic background. Furthermore, the feedback of  $Y_2(x, y, t)$  in  $Y_1(x, y, t)$  allows a stable edge detection through the time, allowing edge



coherence through video WS. In Table 1, on the column of PGSeg, the lowest value of  $\mu FI$  was obtained with LB video. In this video, all methods were affected by false edges caused by the plants, couches and reflections. However, among the methods, PGSeg has better performance because even with illumination changes, edge detection results remain constant. In the SS video, all methods generate false edges because of reflections on the floor and the time and date shown in the display, but PGSeg generates better results since it has a better performance detecting appropriately the edges of the electric stairs. In the case of the CU video, PGSeg has better performance because this method has coherence results in time and the rest of the methods were affected by shadows. In the video PK, all methods were affected by false edges of a wall, but PGseg has better results because generated less noise than the others in areas of the scenario that have a tree and a garden. Results are not showed for videos LB, CU, SS and PK for space reasons.



**Fig. 9.** Edge detection results with frame  $t = 460$  of WS the methods: (a) PGSeg. (b) Canny. (c) LoG. (d) Roberts. (e) Prewitt. (f) Sobel.

## 5 Conclusions

In this paper we propose a Spiking Neural Network known as GSCM, which was applied in a novel method proposed also in this paper to detect edge called PGSeg. GSCM generates pulses from an internal activity that is based on an excitatory input and an inhibitory input. PGSeg is a method inspired in lower layers of the visual cortex and uses the GSCM to generate the edges of objects in the background model of a video sequence with time varying scenario. Results showed that PGSeg has better performance than other edge methods on detecting edges without noise caused by dynamic background, illumination changes, shadows, and reflections.

The parameters of the GSCM are used as constants in PGSeg. Hence, as future work, the GSCM model is going to be modified such that the parameters of the model can be adjusted based on scenario conditions to improve the performance of PGSeg in edge detection or scenario analysis with any method that uses GSCM.

**Acknowledgment.** This research was supported by Fomix CONACYT-Gobierno del Estado de Chihuahua under grant CHIH-2012-C03-193760 and PRODEP ITCHI-PTC-025.

## References

1. Choe, Y., Miikkulainen, R.: Contour integration and segmentation with self-organized lateral connections. *Biol. Cybern.* **90**, 75–88 (2004). Springer
2. Ursino, M., Magosso, E., Cuppini, C.: Recognition of abstract objects via neural oscillators: interaction among topological organization, associative memory and gamma band synchronization. *IEEE Trans. Neural Netw.* **20**(2), 1871–1884 (2009)
3. Yu, G., Slotine, J.-J.: Visual grouping by neural oscillator networks. *IEEE Trans. Neural Netw.* **20**(12), 1871–1884 (2009)
4. Wang, Z., Ma, Y., Cheng, F., Yang, L.: Review of pulse-coupled neural networks. *Image Vis. Comput.* **28**(1), 5–13 (2010)
5. Gu, X.: A new approach to image authentication using local image icon of unit-linking PCNN. In: *International Joint Conference on Neural Networks*, pp 1036–1041 (2006)
6. Wang, Z., Ma, Y.: Medical image fusion using m-PCNN. *Inf. Fusion* **9**(2), 176–185 (2008)
7. Chen, Y., Ma, Y., Kim, H.D., Park, S.K.: Region-based object recognition by color segmentation using a simplified PCNN. *IEEE Trans. Neural Netw. Learn. Syst.* **26**(8), 1682–1697 (2015)
8. Karamiani, A., Farajzadeh, N.: Detecting and tracking moving objects in video sequences using moving edge features. In: *Scientific Cooperations International Workshops on Electrical and Computer Engineering Subfields*, pp 88–92 (2014)
9. Brutzer, S., Höferlin B., Heidemann, G.: Evaluation of background subtraction techniques for video surveillance. In: *Conference on Computer Vision and Pattern Recognition*, pp. 1937–1944 (2011)
10. Miikkulainen, R., Bednar, J.A., Choe, Y., Sirosh, J.: *Computational Maps in the Visual Cortex*, 1st edn. Springer Science Media Inc., New York (2005)
11. Krüger, N., Jansen, P., Kalkan, S., Lappe, M., Leonardi, A., Piater, J., Rodriguez-Sanchez, A., Wiskott, L.: Deep hierarchies in the primary visual cortex: what can we learn for computer vision. *IEEE Trans. Pattern Anal. Mach. Learn.* **35**(8), 1847–1871 (2013)
12. Orozco-Rodriguez, H., Chacon-Murguia, M., Ramirez-Quintana, J.: A neural bioinspired scheme for head pose recognition. In: *Signal Processing and Signal Processing Education Workshop*, pp. 403–408 (2015)
13. Ramirez-Quintana, J., Chacon-Murguia, M.: Self-adaptive SOM-CNN neural system for dynamic object detection in normal and complex scenarios. *Pattern Recogn.* **48**(4), 1137–1149 (2015)
14. Maddalena, L., Petrosino, A.: A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Trans. Image Process.* **17**(7), 168–1177 (2008)
15. Mofrada, M.H., Sadeghi, S., Rezvanian, A., Meybodi, M.R.: Cellular edge detection: combining cellular automata and cellular learning automata. *Int. J. Electron. Commun.* **69**(9), 1282–1290 (2015)
16. Uguz, S., Sahin, U., Sahin, F.: Edge detection with fuzzy cellular automata transition function optimized by PSO. *Comput. Electr. Eng.* **43**(1), 180–192 (2015)
17. Liu, X., Fang, S.: A convenient and robust edge detection method based on ant colony optimization. *Opt. Commun.* **353**(1), 147–157 (2015)
18. Jiang, W., Lam, K.-M., Shen, T.-Z.: Efficient edge detection using simplified Gabor wavelets. *IEEE Trans. Syst. Man Cybern.* **39**(4), 1036–1047 (2009)