


# Computational Mechanisms of Pulse-Coupled Neural Networks: A Comprehensive Review

Kun Zhan<sup>1</sup>  · Jinhui Shi<sup>1</sup> · Haibo Wang<sup>1</sup> · Yuange Xie<sup>1</sup> · Qiaoqiao Li<sup>1</sup>

Received: 19 April 2016 / Accepted: 12 July 2016  
© CIMNE, Barcelona, Spain 2016

**Abstract** Pulse-coupled neural networks (PCNN) have an inherent ability to process the signals associated with the digital visual images because it is inspired from the neuronal activity in the primary visual area, V1, of the neo-cortex. This paper provides insight into the internal operations and behaviors of PCNN, and reveals the way how PCNN achieves good performance in digital image processing. The various properties of PCNN are categorized into a novel three-dimensional taxonomy for image processing mechanisms. The first dimension specifies the time matrix of PCNN, the second dimension captures the firing rate of PCNN, and the third dimension is the synchronization of PCNN. Many examples of processing mechanisms are provided to make it clear and concise.

## 1 Introduction

Synchronization of pulse-coupled biological oscillators in the  $\gamma$  band plays a fundamental role in cortical computation. At the beginning, synchronization of neural activity in the  $\gamma$  band was independently discovered by Eckhorn's group [1] and Singer's group [2] in the primary visual cortex. The discovery of  $\gamma$  band oscillations is considered to be a significant progress in neuroscience, and numerous studies have discussed the processes underlying the  $\gamma$  band oscillations [3–8]. After the discovery of the  $\gamma$  band oscillations, Eckhorn et al. [9] proposed a linking field model with assuming that stimulus-driven feedforward

streams are combined with stimulus-induced feedback streams to enable synchronization. As the assumption is in accordance with the binding feature by synchronization, temporal correlation of the linking field model has the potential capability of segmenting image and obtaining invariant feature [10–12]. Temporal correlation encodes the feature binding among neurons [13–15], and these findings supports the binding by synchronization addressed in special issues of *Neuron* in September 1999 [16] and *Visual Cognition* in June 2001 [17], respectively.

Johnson et al. [18–22] studied on the  $\gamma$  band synchronous spike dynamics and developed pulse-coupled neural networks (PCNN). The ‘pulse-coupled neural network’ appeared firstly in [18, 19], and the classical PCNN was presented in [23–26]. PCNN has been widely applied to image processing [27–31]. There is growing theoretical interest to PCNN spike timing and firing rate as additional variables in the image processing, and the evidence can be seen from special issue of *IEEE Transactions on Neural Networks* on PCNN in May 1999 [32] and on temporal correlation in July 2004 [33].

In the following section, we briefly introduce the recent progress of PCNN in different subjects.

## 2 Recent Progress

PCNN is directly inspired from the  $\gamma$  band synchronization via pulse-coupled synaptic modulations, so it has the inherent ability to segment images even when there is a considerable overlap between adjacent regions [34–50]. Neurons fire synchronously in a region using recruitment based on stimulus similarity and spatial proximity. Using the image segmentation performance of PCNN, it has been widely applied to various fields such as biomedical engineering

---

✉ Kun Zhan  
kzhan@lzu.edu.cn

<sup>1</sup> School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, Gansu, China

[51–60], object detection [61–76], remote sensing [77–81], etc. Klar et al. designed an integrated circuit to speed up the PCNN algorithm [82, 83] and the designed chip is applied to image object separation [83]. Klar et al. proposed a modified PCNN with synaptic plasticity [84, 85], and it is applied to image segmentation [85].

As the invariant feature representation is a crucial component of the binding by synchronization [86–88], the series output pulse images of PCNN can produce invariant feature [20–22]. The PCNN feature is quite robust against geometrical changes and the effect of noise, which is a fundamental property of PCNN. Based on this property, PCNN is applied to texture retrieval [89–93], feature extraction [94–97], and object recognition [98–108].

Time matrix of PCNN records the firing order of neurons and reflects the synchronization [27, 36, 90, 109]. The intensity of a noisy pixel is expected to be significantly different from the intensities of pixels surrounding the noisy pixel [34], so a degraded pixel can be identified from its neighborhood by using time matrix of PCNN [110]. Although the original image restoration PCNN method was proposed by Ranganath et al. [34], until now, the same strategy is still used to restore degraded images [110–120].

Johnson et al. had emphasized the importance of their PCNN factoring model [27, 121–123], and even the corresponding pseudocode is given in [27, 121]. The factoring model can be applied to shadow removal in an image [27, 124], image enhancement [123] and multi-scale analysis [122, 125]. The time matrix of PCNN is applied to image enhancement by using mechanism similar to the factoring model [90, 109].

The pulse firing rate has a non-linear relationship with the intensity of the input images [20–22], and the linking term of PCNN can simulate the pulse firing rate well. Using a multi-channel PCNN, the linking terms from different channels can be integrated to fuse images [126–130]. After these pioneer PCNN fusion methods were proposed [126–130], many algorithms adopt PCNN to fuse images [131–158].

Despite this encouraging progress, there is still little insight into the internal operations and behaviors of the model, or how PCNN achieves such good performance. Without clear understanding of how and why it works, the development of better performances is reduced to trial-and-error.

In this paper, we deeply analyze the computational mechanisms of PCNN to extract its computing characteristics and the essence principles.

### 3 Classical PCNN

There are four main components in PCNN: the dendritic tree, the membrane potential, the dynamic threshold and the action potential. The feeding synapses and the linking

synapses are connected on their dendritic branch. Electrical charges at the synapses produce the membrane potential. If the membrane potential exceeds its threshold, an action potential is generated.

As shown in Fig. 1, the feeding synapses, the linking synapses and the threshold potential are all represented by leaky integrators [9, 19, 23, 24].

#### 3.1 Leaky Integrator

Leaky integrators are the fundamental components of neural networks [159]. The dynamics potential  $v(t)$  of a neural oscillator is described via a leaky integrator,

$$\frac{dv(t)}{dt} = -\alpha v(t) + s \quad (1)$$

where  $t$  is time,  $s$  is the input and  $\alpha$  ( $0 \leq \alpha \leq 1$ ) is the rate of ‘leaking’.

The potential (1) can be discretized as,

$$\frac{V(n) - V(n-1)}{n - (n-1)} = -\alpha V(n-1) + s \quad (2)$$

where  $V(n)$  is the discretized potential, and  $n$  is the discrete time.

We can rewrite (2) as,

$$V(n) = e^{-\alpha_v} V(n-1) + s \quad (3)$$

where  $\alpha_v$  is a positive constant.

#### 3.2 Membrane Potential

A neuron is mostly bidirectionally connected: feedforward and feedback [9, 160–162]. As shown in Fig. 1, the feeding synapses are connected to stimulus that is the main input signal, whereas the linking synapses receive auxiliary signals to modulate the feeding inputs. When a synapse receives postsynaptic action potential from the neighboring neurons, it is charged and decays exponentially.

In classical PCNN [23, 24, 26], the feeding input, the linking input and the membrane potential are given by,

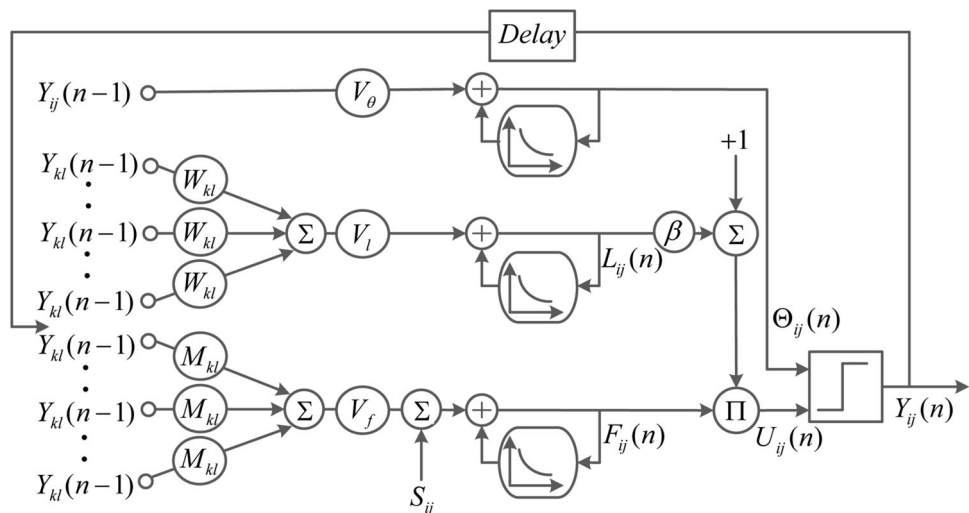
$$F_{ij}(n) = e^{-\alpha_f} F_{ij}(n-1) + V_f \sum_{kl} M_{kl} Y_{kl}(n-1) + S_{ij}, \quad (4)$$

$$L_{ij}(n) = e^{-\alpha_l} L_{ij}(n-1) + V_l \sum_{kl} W_{kl} Y_{kl}(n-1), \quad (5)$$

$$U_{ij}(n) = F_{ij}(n)(1 + \beta L_{ij}(n)) \quad (6)$$

where each neuron is denoted with indices  $(i, j)$ , one of its neighboring neuron is denoted with  $(k, l)$ ,  $M_{kl}$  is the feedforward synaptic weight,  $W_{kl}$  is the linking synaptic weight, and  $\beta$  is a constant amplifying the linking input impact.

**Fig. 1** Schematic of the classical pulse-coupled neural network



### 3.3 Dynamic Threshold

The threshold of PCNN is an evolution from the threshold of the neuron analog in [9, 163]. The absolute and relative refractory period are simulated well by the threshold [163].

In classical PCNN [23–26], the threshold is given by,

$$\Theta_{ij}(n) = e^{-\alpha_\theta} \Theta_{ij}(n-1) + V_\theta Y_{ij}(n-1) \quad (7)$$

where the action potential  $Y_{ij}(n)$  increases the threshold by an amount  $V_{\theta}$  so that a secondary action potential has no capability to be generated during a certain period, and the increased threshold decays with the constant  $e^{-\alpha_{\theta}}$ .

Before a neuron generate the first postsynaptic action potential, the threshold  $\Theta_{ij}(n)$  decreases from the initial threshold  $\Theta_{ij}(0)$  by exponential law with the increase of  $n$ ,  $\Theta_{ij}(n) = g^n \Theta_{ij}(0)$  (8)

where  $e^{-\alpha_\theta}$  is denoted by  $g(0 < g < 1)$  so that some derivations of equations are easy to understand.

Besides the exponential decay, the linear decay was used in PCNN [27],

$$\Theta_{ij}(n) = \Theta_{ij}(n-1) - \delta + V_{\theta} Y_{ij}(n-1) \quad (9)$$

where  $\delta$  is a constant linear decay step.

Hence, the decay function (8) can be replaced with other decay function.

### 3.4 Action Potential

The temporal structure of the action potentials is determined by the membrane potential and the threshold of neurons. When the membrane potential of a neuron exceeds its threshold, the neuron produces an action potential. The action potential  $Y_{ji}(n)$  is given by,

$$Y_{ij}(n) = \begin{cases} 1, & \text{if } U_{ij}(n) > \Theta_{ij}(n), \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

The step activation function (10) is often termed the transfer function in neural network studies.

The activation function of PCNN can be replaced with other frequently used function, such as sigmoid function and radial-basis function.

The input of the activation function is given by,

$$X_{ij}(n) = U_{ij}(n) - \Theta_{ij}(n) \quad (11)$$

then, the sigmoid activation function can be given by,

$$Y_{ij}(n) = \frac{1}{1 + e^{-\gamma_x X_{ij}(n)}} \quad (12)$$

where  $\gamma_x$  is a parameter of the sigmoid function. A sigmoid curve has an ‘S’ shape, with its slope increases as  $\gamma_x$  increases.

The radial-basis activation function can be given by,

$$Y_{ij}(n) = e^{-X_{ij}^2(n)}. \quad (13)$$

### 3.5 PCNN

In order to describe and analyze PCNN easily in the following sections, the classical PCNN is given by Definition 1.

**Definition 1** The classical PCNN is given by (4), (5), (6), (7) and (10).

Different from the conventional integrate and fire model, PCNN has a secondary synapse and the dynamic threshold [27]. The secondary synapse is the linking synapse inspired from  $\gamma$  band synchronization [9], and the

dynamic threshold is designed to simulate the refractory period of a neuron [163].

The classical PCNN has 8 scalar parameters ( $\alpha_f$ ,  $V_f$ ,  $\alpha_l$ ,  $V_l$ ,  $\alpha_\theta$ ,  $V_\theta$ ,  $\beta$  and  $N$ ) and 2 matrix parameters ( $M$  and  $W$ ). There are 3 scalar variables ( $n$ ,  $i$ , and  $j$ ) and 5 matrices variables ( $F_{ij}(n)$ ,  $L_{ij}(n)$ ,  $U_{ij}(n)$ ,  $Y_{ij}(n)$ , and  $\Theta_{ij}(n)$ ).  $n(n \in [0, N])$  is time variable and  $(i, j)$  is spatial coordinate variables.  $F_{ij}(0)$ ,  $L_{ij}(0)$ ,  $\Theta_{ij}(0)$  and  $Y_{ij}(0)$  need to be initialized before the network iteration.

As shown in Fig. 2, the states of  $F_{ij}(n)$ ,  $L_{ij}(n)$ ,  $U_{ij}(n)$ ,  $Y_{ij}(n)$ , and  $\Theta_{ij}(n)$  of a single neuron is iteratively updated due to the increasing of time  $n$ . In the example of Fig. 2, scalar parameters are set as  $\alpha_f = 0.2$ ,  $V_f = 0.1$ ,  $\alpha_l = 0.5$ ,  $V_l = 0.2$ ,  $\beta = 0.5$ ,  $\alpha_\theta = 0.2$ ,  $V_\theta = 6$  and  $N = 40$ , matrix parameters are set as  $M = W = [0.707, 1, 0.707]$ , initial values are set as  $F_{ij}(0) = 0$ ,  $L_{ij}(0) = 0$ ,  $\Theta_{ij}(0) = 1$ , and  $Y_{ij}(0) = 0$ , and the input  $S_{ij} = 0.8$ .

To obtain the central terms of the membrane potential, we assume that the linking strength  $\beta$  is set to 0 and the convolution kernel in the feeding input is also set to 0. Then, the membrane potential is given by,

$$U_{ij}(n) = fU_{ij}(n-1) + S_{ij} \quad (14)$$

where  $e^{-\alpha_f}$  is denoted by  $f(0 < f < 1)$  so that some derivations are easy to understand.

With (14), the function relationship between the membrane potential  $U_{ij}(n)$  and the iteration time  $n$  is expressed by,

$$U_{ij}(n) = \left( U_{ij}(0) - \frac{S_{ij}}{1-f} \right) f^n + \frac{S_{ij}}{1-f}. \quad (15)$$

To explain the mechanism of PCNN clear and concise, we assume that the membrane potential is approximately equal to the stimulus in some following sections. As shown in Fig. 2, the feeding synaptic modulation is the main contribution of the membrane potential (e.g.  $\beta$  is set to a

value that tends to 0) and the main input of the feeding term is the stimulus, i.e., considering (15),  $U_{ij}(n)$  will tend to  $S_{ij}$  if  $f$  tends to zero, or  $U_{ij}(n)$  will tend to  $\frac{S_{ij}}{1-f}$  if  $n$  tends to a large number. As the linking input is a weakly coupled term for the stimulus, the linking strength  $\beta$  is usually set to a small value.  $\beta$  is even set to 0 to simplify PCNN [63, 64, 164].

## 4 Computational Mechanisms

### 4.1 Time Matrix

#### 4.1.1 Time Matrix

A time matrix  $T$  is defined for the first firing time of a neuron [27, 36, 90, 109].

A neuron fires when its membrane potential just exceeds its threshold, i.e., its threshold is lightly smaller than its membrane potential [90, 109]. The threshold is subject to exponential decay, and decays from the first predetermined  $\Theta_{ij}(0)$  by (8). The neuron  $(i, j)$  is firing at the time  $T_{ij}$  when the membrane potential  $U_{ij}(T_{ij})$  exceeds its threshold  $\Theta_{ij}(T_{ij}) = g^{T_{ij}} \Theta_{ij}(0)$ , and the two values are almost same, which is described by,

$$g^{T_{ij}} \Theta_{ij}(0) = U_{ij}(T_{ij}). \quad (16)$$

Hence, the first firing time  $T_{ij}$  is obtained from (16),

$$T_{ij} = \log_g \frac{U_{ij}(T_{ij})}{\Theta_{ij}(0)}. \quad (17)$$

(17) is described to represent the analytical solution of the time matrix. As (17) is an implicit function, the time matrix can not be obtained by (17). In implementation, we obtain the time matrix by Definition 2.

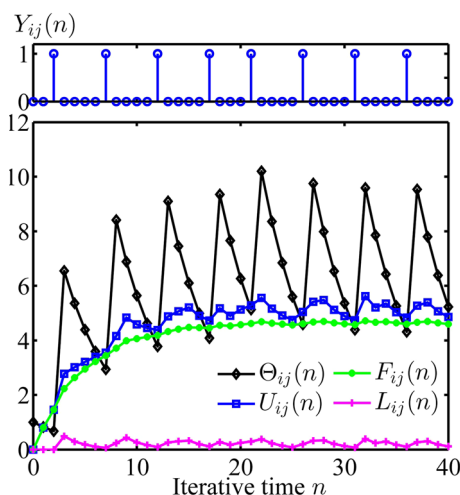
**Definition 2** The time matrix  $T_{ij}$  is obtained on the premise that the threshold amplification factor  $V_\theta$  is set to a large enough value so it makes sure that neurons fire only once,

$$T_{ij}(n) = T_{ij}(n-1) + nY_{ij}(n). \quad (18)$$

#### 4.1.2 Premise of Time Matrix

This section shows how to select  $V_\theta$  to guarantee each neuron pulses exactly once during a pulsing cycle.

We assume that the membrane potential is equal to the stimulus  $S_{ij} \in [0, 1]$  and the initial threshold is  $\Theta(0) = 1$ . The neuron located at  $x$  with the maximum stimulus fire at the first time  $T_x$ . Considering (17), if the thresholds decay by  $g$  exponentially,  $T_x$  is given by,



**Fig. 2** Neural activity of a single neuron

$$T_x = \log_g \frac{S_x}{\Theta_x(0)} = \log_g S_x. \quad (19)$$

Similarly,

$$T_y = \log_g S_y \quad (20)$$

where  $S_y$  denotes the minimum stimulus, and  $T_y$  denotes the firing time of neuron located at  $y$  whose stimulus is  $S_y$ .

At time  $T_y$ , the threshold at location  $x$  is

$$\begin{aligned} \Theta_x(T_y) &= \Theta_x(T_x + 1)g^{T_y - (T_x + 1)} \\ &= (g^{T_x + 1}\Theta_x(0) + V_\theta)g^{T_y - T_x - 1} \\ &= g^{T_y} + V_\theta g^{T_y - T_x - 1}. \end{aligned} \quad (21)$$

Each neuron is guaranteed to fire only once during a cycle when  $V_\theta$  is selected to promise that  $\Theta_x(T_y)$  is not smaller than the maximum possible value of  $S_x$ . Hence,

$$g^{T_y} + V_\theta g^{T_y - T_x - 1} > S_x. \quad (22)$$

Substituting (19) and (20) into (22),

$$V_\theta > g \frac{S_x}{S_y} (S_x - S_y). \quad (23)$$

If thresholds decay by  $\delta$  linearly, the firing time is

$$T_x = \frac{\Theta_x(0) - S_x}{\delta} = \frac{1 - S_x}{\delta}, \quad (24)$$

$$T_y = \frac{\Theta_y(0) - S_y}{\delta} = \frac{1 - S_y}{\delta}. \quad (25)$$

At time  $T_y$ , the threshold at location  $x$  is

$$\Theta_x(T_y) = V_\theta + 1 - \delta T_y. \quad (26)$$

As previous stated,  $V_\theta$  should meet the following requirement,

$$V_\theta + 1 - \delta T_y > S_x. \quad (27)$$

Substituting (25) into (27),

$$V_\theta > S_x - S_y. \quad (28)$$

In practice, we can estimate the value of  $V_\theta$  by using (23) and (28) to obtain the time matrix [27, 35, 48].

#### 4.1.3 Single-Pass

The single-pass form can be used to obtain the time matrix. The single-pass, a working form of PCNN, is completed when all of neurons generate the action potentials [27]. The single-pass can be realized when the threshold amplification factor  $V_\theta$  is set to a large enough value so it makes sure that neurons are generated the action potential only once [36, 90].

The single-pass working form of PCNN is a method for the network stopping condition. Different from the

manually setting the network iterative times, the iterative process is automatic stopping in the single-pass form. The stopping condition is that all neurons have fired.

In implementation, it can be given by Algorithm 1. In Algorithm 1,  $\mathcal{N}$  is the total number of neurons, and *firednum* is counting the number of the fired neurons.

---

#### Algorithm 1 The single-pass algorithm

---

```

1: firednum = 0, and  $n = 0$ .
2: while firednum <  $\mathcal{N}$  do
3:    $n = n + 1$ .
4:   Update  $F$ ,  $L$ ,  $U$ ,  $\Theta$ , and  $Y$  by (4), (5), (6), (7) and (10), respectively.
5:   firednum = firednum + sum( $Y$ ).
6:    $T = T + nY$ .
7: end while
    
```

---

## 4.2 Firing Rate

### 4.2.1 Firing Rate

We assume that the membrane potential is equal to the stimulus because of the reason as discussed at the end of Sect. 3.5.

The firing condition is the membrane potential is larger than the decreasing threshold. The precise firing time is when the membrane potential is almost equal to the threshold,

$$\Theta_{ij}(n) = S_{ij}. \quad (29)$$

At the next iteration of PCNN, the threshold is changed to,

$$\Theta_{ij}(n + 1) = g\Theta_{ij}(n) + V_\theta = gS_{ij} + V_\theta \quad (30)$$

then the threshold delays with exponential law.

The threshold delays from the initial value  $\Theta_{ij}(0)$  at first firing time, it delays from  $gS_{ij} + V_\theta$  at subsequent firing time, and the firing time is when the threshold is equal to  $S_{ij}$ . Therefore, the action potential of a neuron occurs at the iteration temporal index  $n$ ,

$$n = \log_g \frac{S_{ij}}{\Theta_{ij}(0)} + \sum_{n > 1} \log_g \frac{S_{ij}}{gS_{ij} + V_\theta} \quad (31)$$

The firing time (31) is under the assumption that the membrane potential is a constant for a single neuron, and it is only an approximate analytical solution. In [21, 27, 34, 90], the firing frequency  $\mathcal{F}_{ij}$  of a neuron is given by,

$$\mathcal{F}_{ij} = \frac{1}{\log_g \frac{S_{ij}}{gS_{ij} + V_\theta}} \quad (32)$$



A neuron with strong stimulus is generally fired earlier than a neuron with weak stimulus. If neurons with the weakest stimulus are fired, it implies an end of a complete cycle. According to Definition 3, we can obtain the number of firing times in the cycle.

**Definition 3** The firing rate  $\mathcal{F}_{ij}$  of a neuron with initial values of 0's is given by,

$$\mathcal{F}_{ij}(n) = \mathcal{F}_{ij}(n-1) + Y_{ij}(n) \quad (33)$$

where  $\mathcal{F}_{ij}$  counts the firing times of a neuron at  $(i, j)$  after the network iterates  $N$  times.

#### 4.2.2 Multiple-Pass

The multiple-pass working form of PCNN is conducted without a prerequisite, and it also a method for the network stopping condition that the network stops iteration when all neurons are fired. it can be used to obtain the firing rate of PCNN.

In implementation, it can be given by Algorithm 2. In Algorithm 2, the  $Y_a$  with initial values of 0's has the same size of  $Y$ , and the function  $\text{or}(\cdot, \cdot)$  is the point-wise 'or' operation.

---

#### Algorithm 2 The multiple-pass algorithm

---

```

1:  $firednum = 0$ .
2: while  $firednum < \mathcal{N}$  do
3:   Update  $F$ ,  $L$ ,  $U$ ,  $\Theta$ , and  $Y$  by (4), (5), (6), (7) and
     (10), respectively.
4:    $Y_a = \text{or}(Y_a, Y)$ .
5:    $firednum = \text{sum}(Y_a)$ .
6:    $\mathcal{F} = \mathcal{F} + Y$ .
7: end while

```

---

### 4.3 Synchronization

#### 4.3.1 Feeding Wave

Waves in PCNN are studied at the beginning of the model being proposed [18, 19]. Changes in the action potential matrix  $Y$  are observed, and the changes look like waves traveling.

We define that the waves related with the feeding synapse are the feeding wave. We explore the concept in the context of a specific example. The specific example of the feeding wave can be described as,

$$\begin{cases} U_{ij}(n) = 4 \sum_{kl} Y_{kl}(n-1). \\ Y_{ij}(n) = \begin{cases} 1, & \text{if } U_{ij}(n) > 3, \\ 0, & \text{otherwise.} \end{cases} \end{cases} \quad (34)$$

The initial  $Y(0)$  is shown in Fig. 3a where each block means a single pixel and the blank white blocks correspond to 0 in intensity, and the iterative number  $N$  is set to 5. The feeding wave can be observed from the first row of Fig. 4.

The feeding wave plays an important role in PCNNs. During (34) iteration, centrifugal wave emanates outward. The feeding wave travels forward step by step along the boundary. As the feeding wave is same to the dilation operation for binary image, many thinning algorithms are developed based on the wave [165–168]

#### 4.3.2 Linking Wave

The linking wave is related to the linking synaptic input terms in PCNN, and a good application of the linking wave is finding the shortest path in a maze image [169–171]. An example of the linking wave can be described as,

$$\begin{cases} U_{ij}(n) = S_{ij} \left( 1 + \beta \sum_{kl} Y_{kl}(n-1) \right). \\ Y_{ij}(n) = \begin{cases} 1, & \text{if } U_{ij}(n) > 3, \\ 0, & \text{otherwise.} \end{cases} \end{cases} \quad (35)$$

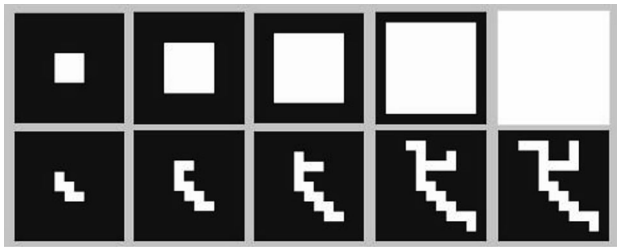
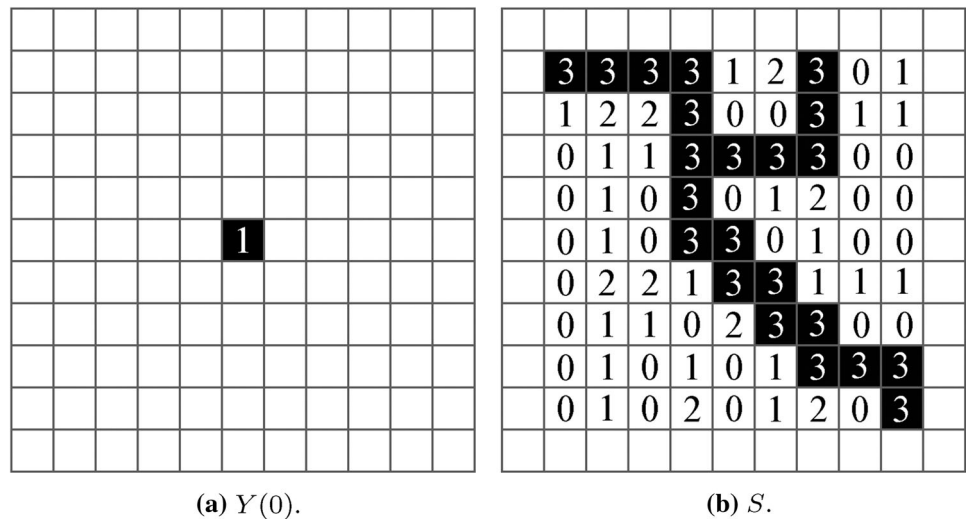
The linking wave is shown in the second row of Fig. 4. In the example,  $\beta$  is set to 0.1, the initial  $Y(0)$  is shown in Fig. 3a and the stimuli is shown in Fig. 3b where the numbers within blocks mark the intensity values.

During (35) iteration, the linking wave travels forward along the stimuli similarity and the spatial proximity. The area where the linking wave can reach belongs to the regions with the stimuli similarity and the spatial proximity.

#### 4.3.3 Synchronization

PCNN are inspired from the  $\gamma$  band synchronization. There are two distinct types of synchronization in PCNN, and they are stimulus-forced and stimulus-induced [9, 109]. The stimulus-forced synchronization is related with the feeding wave, and the stimulus-induced synchronization is related with the linking wave. The two forms of waves spread from a central neuron roundly while radii of wave increase step by step. All the neurons in the neighborhood are coupled by each other, and a fired neuron captures some of neighboring neurons to fire synchronously. Several neighboring neurons are fired and their neighboring can be captured by them. The feeding wave propagates to its neighboring neurons once a neuron fires previously, and the linking wave only select neurons whose stimuli are similar to the previous fired neuron.

**Fig. 3** Initial pulse and stimuli matrix



**Fig. 4** The first row shows the feeding waves and the second row shows the linking waves

#### 4.3.4 Fast-Linking

The fast-linking means that neurons with similar stimuli become temporal correlated and fire synchronously fast than normal [27, 36, 172]. In temporal correlated theory, time plays the role of binding: different stimuli unfold in time [173].

There are two loops in the fast-linking. The feeding input, the linking input, the membrane potential and the action potential terms are iterated in internal loop until the action potential term is not varying any more, and the threshold is iterated in external loop. Algorithm 3 shows the fast-linking working form. In Algorithm 3, the function  $\text{equal}(\text{post\_}Y, Y)$  returns logical 1 if arrays  $\text{post\_}Y$  and

$Y$  are the same size and contain the same values, and logical 0 otherwise.

#### Algorithm 3 The fast-linking algorithm

```

1:  $n = 0$ .
2: while  $n < N$  do
3:    $n = n + 1$ .
4:   Update  $\Theta$  by (7).
5:   repeat
6:      $\text{post\_}Y = Y$ .
7:     Update  $F$ ,  $L$ ,  $U$ , and  $Y$  by (4), (5), (6) and (10), respectively.
8:   until  $\text{equal}(\text{post\_}Y, Y)$ .
9: end while
    
```

#### 4.3.5 Inhibition-Linking

We wish to achieve synchronization in homogeneous stimuli and de-synchronization between heterogeneous stimuli. A global negative inhibitory term is designed to support the de-synchronization [10, 11, 35, 36, 109, 174–176].

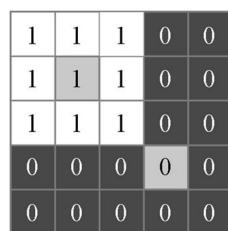
A simple example of the inhibition-linking mechanism can be explained by a modified linking synaptic term,

$$L_{ij}(n) = \sum_{kl} Y_{kl}(n-1) - d \quad (36)$$

where  $d$  is a positive constant for the global inhibitory.

We assume that the action potential matrix  $Y_{ij}(n-1)$  has the values as shown in Fig. 5 at a time point. There are two neurons depicted by the gray color, and they belong to a white region (i.e. values are 1) and a black region (i.e. values are 0), respectively. If  $d$  is set to different values, (36) will produce different values for the two gray neurons [36]. If  $d$  is set to 0, the ratio of linking input received by

**Fig. 5** An action potential matrix  $Y_{ij}(n-1)$  at a time point



the two gray neurons is 9:1. If  $d$  is set to 0.5, it will rise to a ratio of 17:1.

## 5 Image Processing Mechanisms

There is a corresponding neuron for each pixel in an image, and the stimulus corresponds to the grayscale intensity of the image. The neuron is located at the pixel  $(i, j)$ , and the input image is a matrix with resolution  $r \times c$ .

### 5.1 Histogram

As the image histogram plays an important role in image processing, the relationship between histogram and PCNN is analyzed in this section. It is conducive to understanding the mechanisms of how to process image with PCNN.

**Theorem 1** Suppose an image  $S \in [0, 255]$  has 256 gray levels and the membrane potential is equal to stimulus. The single-pass working form of PCNN is adopted and the threshold (9) decays from  $\Theta_{ij}(0) = 255$  by  $\delta = 1$  linearly, then the histogram can be obtained by

$$H(256 - n) = \sum_{ij} Y_{ij}(n), \quad n \in [0, 255]. \quad (37)$$

*Proof* At the first iteration  $n = 1^{\text{st}}$ , the threshold turns to  $\Theta(1) = 254$ , so the neurons with intensity of 255 fire, and  $\sum_{ij} Y_{ij}(1)$  returns the number of neurons whose stimulus is 255. Hence,

$$H(255) = \sum_{ij} Y_{ij}(1). \quad (38)$$

At the following iteration, because of the single-pass, the number of “1” in  $Y(n)$  returns the number of neurons whose stimulus is exactly equal to  $256 - n$ , which can be expressed by (37).  $\square$

#### Algorithm 4 Obtain image histogram using PCNN

```

1: function HISTOGRAM( $S$ )
2:   Initialize:  $\Theta = 255, \delta = 1$ , and  $V_\theta = 256$ .
3:   for  $n \in [0, 255]$  do
4:      $U = S$ .
5:      $\Theta = \Theta - \delta + V_\theta Y$ .
6:      $Y = \text{step}(U - \Theta)$ .
7:      $H(256 - n) = \text{sum}(Y)$ .
8:   end for
9:   return  $H$ .
10: end function

```

Algorithm 4 shows a function to obtain histogram by a simplified PCNN. Fig. 6a shows the image Lena, and Fig. 6b shows its histogram. In Fig. 6b, the blue dotted line represents the histogram obtained by the standard method,

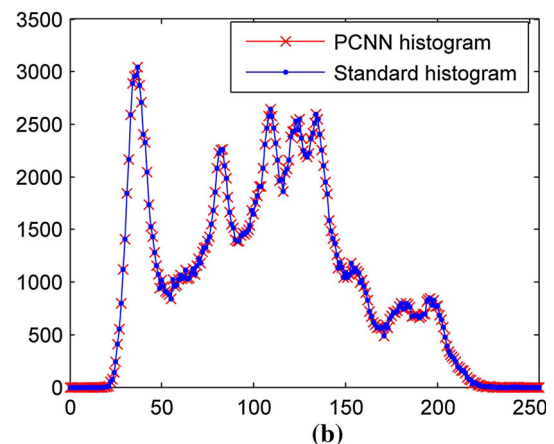
and the red cross line represents the histogram obtained by using Algorithm 4. They are exactly same to each other.

### 5.2 Image Segmentation

We wish to segment image to achieve synchronization in each region and de-synchronization between adjacent regions. Image segmentation based on temporal correlation mainly depends on lateral potential modulation, and linking wave ensures synchronization among pixels responding to the same region. Synchronization of a connected set of neurons corresponds to region labeling, and different regions are given distinct labels in independent differing times. Temporal correlation of the output spike provides an image segmentation property, and one neuron can fire and cause a domino effect that continues until all neurons with the intensity similarity and the spatial proximity fire in phase synchrony. Phase locking is mediated by the stimulus similarity and the spatial proximity, and neurons are



(a)



(b)

**Fig. 6** An image Lena and its histogram. **a** The image Lena, **b** the histogram of the image Lena



fired synchronously in a region using recruitment. Segments of the image consisting of pixels to similar intensity values tend to spike synchronously. Same object becomes temporal correlated and distinct objects become anti-correlated. The method is able to identify a single connected region with the time matrix in each time step.

---

**Algorithm 5** An example of image segmentation
 

---

```

1: Input:  $S$  is assigned by values as shown in Fig.7(a).
2: Output:  $O = 256 - T$ .
3: Initialize:  $W$  is assigned to a Gaussian kernel with the
   standard deviation of 1,  $N$  is assigned to the total element
   number of  $S$ ,  $\Theta(0) = 255$ ,  $Y(0) = 0$ ,  $\delta = 1$ ,  $\beta = 2$ ,  $V_\theta =$ 
   400,  $firednum = 0$ , and  $n = 0$ .
4: while  $firednum < N$  do
5:    $n = n + 1$ .
6:    $L = \text{sum}(W_{kl}Y_{kl})$ .
7:    $\Theta = \Theta - \delta + V_\theta Y$ .
8:    $Flag = 1$ .
9:   while  $Flag = 1$  do
10:     $Q = Y$ .
11:     $U = S(1 + \beta L)$ .
12:     $Y = \text{step}(U - \Theta)$ .
13:    if  $\text{equal}(Q, Y)$  then
14:       $Flag = 0$ .
15:    else
16:       $L = \text{sum}(W_{kl}Y_{kl})$ .
17:    end if
18:  end while
19:   $firednum = firednum + \text{sum}(Y)$ .
20:   $T = T + nY$ .
21: end while

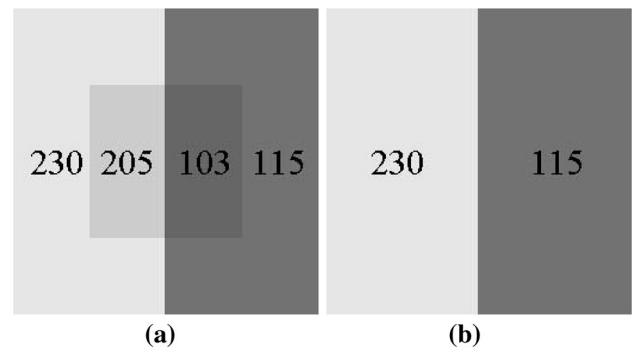
```

---

As shown in Algorithm 5, we combine the single-pass to the fast-linking for image segmentation. The input of the network is shown in Fig. 7a, and the output segmented image is shown in Fig. 7b. At the iterative time  $n = 26$ th, the threshold decays to  $\Theta = 255 - 26 = 229$ . The neurons with  $S = 230$  fire and the neurons with  $S = 205$  can be captured by the waves. Then, these neurons at the white left side of the input are marked in the time matrix with the value 26. At the iterative time  $n = 141$ st, the threshold decays to  $\Theta = 255 - 141 = 114$ . The neurons with  $S = 115$  fire and the neurons with  $S = 103$  can be captured by the waves. Then, these neurons at the black right side of the input are marked in the time matrix with the value 141. The output image  $O$  is obtained by using an inverse operation  $O = 256 - T$ , then the input image is perfectly segmented by PCNN.

### 5.3 Invariant Feature Extraction

We have an intuitive understanding that the histogram of an image is invariant to rotation, shift, scale and other geometrical changes, especially if the image is a texture. As shown in Sect. 5.1, a simplified PCNN can be used to



**Fig. 7** An example of PCNN image segmentation. **a** Input, **b** output

obtain the histogram of an image, so the way we obtain the histogram is defined to the time series,

$$G(n) = \sum_{ij} Y_{ij}(n) \quad (39)$$

where  $G(n)$  denotes the time series and is a 1-D feature.

The time series of an image is not equal to its histogram, but it is more robust against geometrical changes and the effect of noise. Image histogram is generally a  $256 \times 1$  vector, but the time series is a  $N \times 1$  vector (e.g.  $N$  is set to 37 in [89–93]). In Sect. 5.1, histogram is obtained by a simplified PCNN, but the parameters  $g$  or  $\delta$  for obtaining the time series are not set as in Sect. 5.1 generally.

On the other hand, PCNN has the most important image segmentation performance. e.g., if the time series is obtained by Algorithm 5, the output shown in Fig. 7b will be robust against some degree changes of the center square in Fig. 7a.

Hence, for an input image  $S$ , a series of binary images  $Y_{ij}(1), Y_{ij}(2), \dots, Y_{ij}(N)$  are produced, and these binary images can be computed to a unique feature.

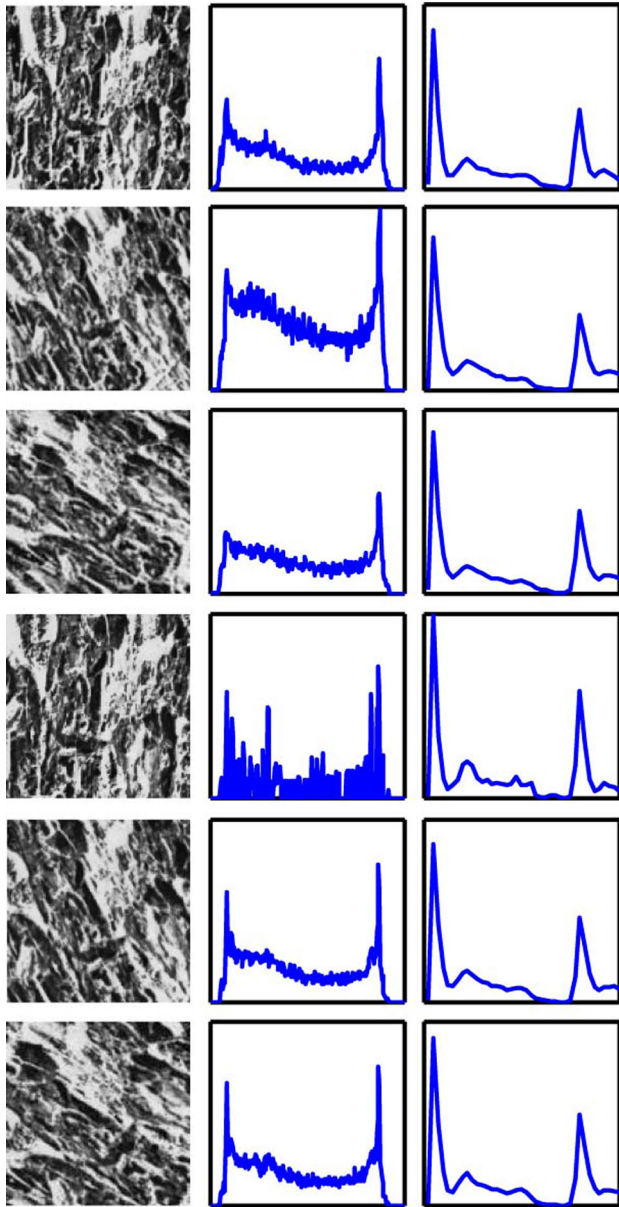
Figure 8 shows that the time series is robust against geometrical changes. Each row of the third column are the time series which are more invariant to geometrical changes of a texture than the corresponding histogram.

### 5.4 Image Restoration

The intensity of a noisy pixel is quite different from the intensity of neighboring pixels. Noise detection is to find the locations of noisy pixels in the image. Then, adjusting the intensities of noisy pixels is called image restoration.

A mean filter can smooth local variations in a degraded image and noise is reduced [178]. PCNN is used to reduce noise in an image because of the pulse synchronization performance.

Noisy pixels always can not synchronize with their neighboring pixels. Neighboring pixels in an image are strongly correlated but noisy pixels are anti-correlated.



**Fig. 8** The *first column* shows an input image (D12 of Brodatz album [177]) and its rotated and resized images, the *second column* shows image histogram and the *third column* shows the time series

This prior knowledge can be captured using the time matrix of PCNN. In a  $3 \times 3$  square neighborhood, if the center element of time matrix is lower than its neighboring elements, which means that the corresponding pixel has a larger intensity, the corresponding intensity is decreased by a small positive constant. If the center is higher than its neighboring elements, the corresponding intensity is increased by the constant.

It has been proved that image restoration using Algorithm 6 can obtain a higher peak-signal-to-noise-ratio than

mean and median filters [90, 110, 179]. In Algorithm 6, the function ‘ $\text{med}(\cdot)$ ’ returns the median value.

---

#### Algorithm 6 PCNN image restoration algorithm

---

```

1: Input: A noised image,  $S \in \mathbb{R}^{r \times c}$ ,  $S_{ij} \in [0, 1]$  with Gaussian noise of  $\mu = 0$  and  $\sigma = 0.001$  [90].
2: Output: Image  $O$  with noise removed.
3: Initialize:  $\Delta = 0.02$ .
4: Obtain the time matrix  $T$  by Algorithm 1
5: for  $i \in [1, r], j \in [1, c]$  do
6:    $K = T_{i-1:i+1, j-1:j+1}$ 
7:   if  $\text{equal}(T_{i,j}, \text{med}(K))$  then
8:      $O_{i,j} = S_{i,j}$ 
9:   else if  $\text{equal}(T_{i,j}, \max(K))$  then
10:     $O_{i,j} = S_{i,j} + \Delta$ 
11:   else if  $\text{equal}(T_{i,j}, \min(K))$  then
12:     $O_{i,j} = S_{i,j} - \Delta$ 
13:   else
14:     $O_{i,j} = \text{med}(S_{i-1:i+1, j-1:j+1})$ 
15:   end if
16: end for

```

---

### 5.5 Image Enhancement

How an image is represented can greatly affect how easy it is to do different applications with it [180]. The time matrix of PCNN can be used to represent image [90, 109].

Suppose that the membrane potential is equal to the stimulus, we can rewrite (17),

$$T_{ij} = \log_g S_{ij} - \log_g \Theta_{ij}(0). \quad (40)$$

Corresponding to the Weber–Fechner’s law, it can be seen from (40) that the time matrix has an approximate logarithmal relation with the stimuli matrix, which is consistent with Weber–Fechner’s law.

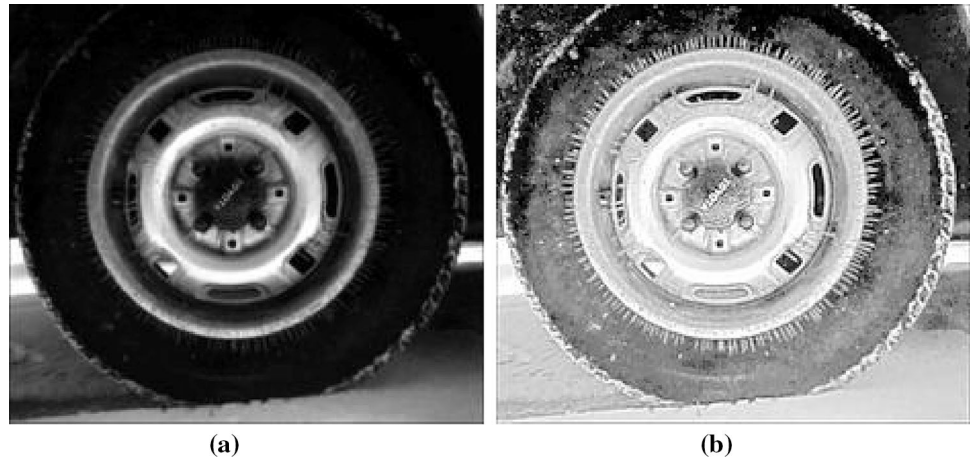
The image enhancement algorithm can be achieved by Algorithm 1. Because a neuron with larger stimulus fires earlier than a neuron with smaller stimulus, the output result needs an inverse operation.

Figure 9 shows results of image enhancement. The output, Fig. 9b, boosts the contrast which are too dark to distinguish clearly in input image while preserves the information of the input image.

### 5.6 Image Fusion

It can be seen from (32) that the firing rate has a non-linear relation with the stimulus. On the other hand, (5) and (33) have similar characteristic if the parameters of the linking term are set properly. Hence, different modal images are processed by different channels of PCNN separately and their linking can be fused by another additional PCNN channel. The details of the additional PCNN fusion channel is discussed in Sect. 6.5.

**Fig. 9** PCNN image enhancement. **a** Input, **b** output



## 6 Modified PCNNs

A problem of PCNN is that it is a complex model with 10 parameters and 8 variables, and one solution of the problem is to modify PCNN.

### 6.1 Intersecting Cortical Model

If  $\beta$  is set to 0, PCNN becomes to an intersecting cortical model [63, 64, 164]. This model only has the feeding wave in its neural activity. The intersecting cortical model is given by,

$$F_{ij}(n) = fF_{ij}(n-1) + \sum_{kl} M_{kl}Y_{kl}(n-1) + S_{ij}.$$

$$\Theta_{ij}(n) = g\Theta_{ij}(n-1) + V_{\theta}Y_{ij}(n-1).$$

$$Y_{ij}(n) = \begin{cases} 1, & \text{if } F_{ij}(n) > \Theta_{ij}(n), \\ 0, & \text{otherwise.} \end{cases}$$

### 6.2 Spiking Cortical Model

Under the consideration that the membrane potential of most biological neural networks is represented by a leaky integrator [159], In spiking cortical model, the membrane potential is simplified to a single equation rather than using three in PCNN. On the other hand, Johoson et al. set the feeding input to the stimulus  $S$  and the linking input to the term  $\sum_{kl} W_{kl}Y_{kl}(n-1)$  in [27, 121]. These studies are the reasons why the spiking cortical model was proposed [90]. The spiking cortical model is given by,

$$U_{ij}(n) = fU_{ij}(n-1) + S_{ij} \sum_{kl} W_{kl}Y_{kl}(n-1) + S_{ij}.$$

$$\Theta_{ij}(n) = g\Theta_{ij}(n-1) + V_{\theta}Y_{ij}(n-1).$$

$$Y_{ij}(n) = \begin{cases} 1, & \text{if } U_{ij}(n) > \Theta_{ij}(n), \\ 0, & \text{otherwise.} \end{cases}$$

### 6.3 Sigmoidal-Linking Model

**Definition 4** The sigmoidal-linking input  $L'_{ij}(n)$  is defined that the original linking term  $L_{ij}(n)$  is the input of a sigmoid function,

$$L'_{ij}(n) = \frac{1}{1 + e^{-\gamma_l(L_{ij}(n)-b)}} \quad (41)$$

where  $\gamma_l$  is a parameter of the sigmoid function, and  $b$  is a constant.

In [27, 30], they let  $\gamma_l$  infinite and turn the sigmoid function into the step function,

$$L'_{ij}(n) = \begin{cases} 1, & \text{if } L_{ij}(n) > b, \\ 0, & \text{otherwise,} \end{cases} \quad (42)$$

which is named as sigmoidal-linking in [27], quantized-linking in [30], and unit-linking in [124].

The sigmoidal-linking renders the network be simpler than the classical PCNN [27, 30, 124] because it is usually given by,

$$F_{ij}(n) = S_{ij}.$$

$$L_{ij}(n) = \begin{cases} 1, & \text{if } \sum_{kl} Y_{kl}(n-1) > 0, \\ 0, & \text{otherwise.} \end{cases}$$

$$U_{ij}(n) = F_{ij}(n)(1 + \beta L_{ij}(n)).$$

$$\Theta_{ij}(n) = \Theta_{ij}(n-1) - \delta + V_{\theta}Y_{ij}(n-1).$$

$$Y_{ij}(n) = \begin{cases} 1, & \text{if } U_{ij}(n) > \Theta_{ij}(n), \\ 0, & \text{otherwise.} \end{cases}$$

### 6.4 Feature-Linking Model

The intersecting cortical model only has the feeding wave, and the spiking cortical and sigmoidal-linking models only have the linking wave. The feature-linking model not only

reduces the parameters but also two types of waves are contained in network. In addition, there is a inhibition-linking term in the feature-linking model. Hence, the feature-linking model [109] is given by,

$$U_{ij}(n) = fU_{ij}(n-1) + \left( \sum_{kl} M_{kl} Y_{kl}(n-1) + S_{ij} \right) \times \left( 1 + \beta \left( \sum_{kl} W_{kl} Y_{kl}(n-1) - d \right) \right).$$

$$\Theta_{ij}(n) = g\Theta_{ij}(n-1) + V_{\theta} Y_{ij}(n-1).$$

$$Y_{ij}(n) = \begin{cases} 1, & \text{if } U_{ij}(n) > \Theta_{ij}(n), \\ 0, & \text{otherwise.} \end{cases}$$

## 6.5 Multiple-Linking Model

The membrane potential  $U_{ij}(n)$  of a neuron is viewed as a general higher-order network containing the multiple-linking modulation [9, 27]. The membrane potential of the multiple-linking model is given by,

$$U_{ij}(n) = F_{ij}(n) \prod_{p=1}^P \left( 1 + \beta^p L_{ij}^p(n) \right) \quad (43)$$

where  $p$  is the index of the linking term and  $P$  is the number of the linking terms.

---

### Algorithm 7 The multiple-linking algorithm

---

```

1: for  $n \in [1, N]$  do
2:   for  $p \in [1, P]$  do
3:     Update  $F^p, L^p, U^p, \Theta^p$ , and  $Y^p$  by (4), (5), (6),
       (7) and (10), respectively.
4:   end for
5:   Update  $F, U, \Theta$  and  $Y$  by (4), (43), (7) and (10),
       respectively.
6: end for
```

---

Algorithm 7 shows how to implement the multiple-linking model. In image fusion, the feeding input  $F_{ij}(n)$  is set to 1 and fusion result is obtained by firing rate of neurons [126–130].

## 7 Conclusion

PCNN is inspired from the  $\gamma$  band oscillations, so it has the inherent ability of image processing, such as image segmentation, image restoration, invariant image feature extraction, image enhancement, image fusion etc. In this paper, we deeply analyze the internal operations and behaviors of PCNN. There are two network-stopping conditions in PCNN: the single-pass and the multiple-pass. The two network-stopping methods are used to obtain the time

matrix and the firing rate. Synchronization of PCNN is related to two types of waves: the feeding wave and the linking wave. The two waves render it easy to understand the behaviors of PCNN. Based on these internal behaviors, we further reveals the image processing mechanisms of PCNN.

**Funding** This study is funded by the National Science Foundation of China under the Grant No. 61201422 and the Specialized Research Fund for the Doctoral Program of Higher Education under the Grant No. 20120211120013.

### Compliance with Ethical Standards

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethical Statement** This study does not involve human participants or animals

## References

- Eckhorn R, Bauer R, Jordan W, Brosch M, Kruse W, Munk M, Reitboeck HJ (1988) Coherent oscillations: a mechanism of feature linking in the visual cortex? *Biol Cybern* 60(2):121–130
- Gray CM, König P, Engel AK, Singer W (1989) Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature* 338(6213):334–337
- Fries P, Nikolić D, Singer W (2007) The gamma cycle. *Trends Neurosci* 30(7):309–316
- Fries P (2009) Neuronal gamma-band synchronization as a fundamental process in cortical computation. *Annu Rev Neurosci* 32:209–224
- Buzsáki G, Wang X-J (2012) Mechanisms of gamma oscillations. *Annu Rev Neurosci* 35:203–225
- Nikolić D, Fries P, Singer W (2013) Gamma oscillations: precise temporal coordination without a metronome. *Trends Cogn Sci* 17(2):54–55
- Brunet N, Vinck M, Bosman CA, Singer W, Fries P (2014) Gamma or no gamma, that is the question. *Trends Cogn Sci* 18(10):507–509
- Brunet NM, Bosman CA, Vinck M, Roberts M, Oostenveld R, Desimone R, De Weerd P, Fries P (2014) Stimulus repetition modulates gamma-band synchronization in primate visual cortex. *Proc Natl Acad Sci* 111(9):3626–3631
- Eckhorn R, Reitboeck HJ, Arndt M, Dicke P (1990) Feature linking via synchronization among distributed assemblies: simulations of results from cat visual cortex. *Neural Comput* 2(3):293–307
- Reitboeck HJ, Stoecker M, Hahn C (1993) Object separation in dynamic neural networks. *IEEE Proc ICNN* 2:638–641
- Stoecker M, Reitboeck HJ, Eckhorn R (1996) A neural network for scene segmentation by temporal coding. *Neurocomputing* 11(2–4):123–134
- Stoecker M, Eckhorn R, Reitboeck HJ (1997) Size and position invariant visual representation supports retinotopic maps via selective backward paths: A dynamic second order neural network model for a possible functional role of recurrent connections in the visual cortex. *Neurocomputing* 17(2):111–132
- Milner PM (1974) A model for visual shape recognition. *Psychol Rev* 81(6):521–535
- von der Malsburg C (1994) The correlation theory of brain function. Springer, Berlin



15. Gray CM (1999) The temporal correlation hypothesis of visual feature integration: still alive and well. *Neuron* 24(1):31–47
16. Roskies AL (1999) The binding problem. *Neuron* 24(1):7–9
17. Müller HJ, Elliott MA, Herrmann CS, Mecklinger A (2001) Neural binding of space and time: an introduction. *Vis Cogn* 8(3–5):273–285
18. Johnson JL (1993) Waves in pulse-coupled neural networks. *Proc World Congr Neural Netw* 4:299–302
19. Johnson JL, Ritter D (1993) Observation of periodic waves in a pulse-coupled neural network. *Opt Lett* 18(15):1253–1255
20. Johnson JL (1994) Pulse-coupled neural networks. *Proc Adapt Comput Math Electron Opt* CR55:47–76
21. Johnson JL (1994) Pulse-coupled neural nets: translation, rotation, scale, distortion, and intensity signal invariance for images. *Appl Opt* 33(26):6239–6253
22. Johnson JL (1994) Time signatures of images. *IEEE Proc ICNN* 2:1279–1284
23. Kinser JM, Johnson JL (1996) Object isolation. *Opt Mem Neural Netw* 5:137–146
24. Kinser JM, Johnson JL (1996) Stabilized input with a feedback pulse-coupled neural network. *Opt Eng* 35(8):2158–2161
25. Kinser JM (1996) Simplified pulse-coupled neural network. *Proc SPIE* 2760:563–567
26. Lindblad T, Becanovic V, Lindsey CS, Szekely G (1997) Intelligent detectors modelled from the cat's eye. *Nuclear Instrum Methods Phys Res A* 389(1):245–250
27. Johnson JL, Padgett ML (1999) PCNN models and applications. *IEEE Trans Neural Netw* 10(3):480–498
28. Ma Y, Li L, Zhan K, Wang Z (2008) Pulse coupled neural network and digital image processing. Science Press, Beijing
29. Ma Y, Zhan K, Wang Z (2011) Applications of pulse-coupled neural networks. Springer, Berlin
30. Lindblad T, Kinser JM (2013) Image processing using pulse-coupled neural networks: applications in python. Springer, Berlin
31. Subashini MM, Sahoo SK (2014) Pulse coupled neural networks and its applications. *Expert Syst Appl* 41(8):3965–3974
32. Johnson JL, Padgett ML, Omidvar O (1999) Guest editorial overview of pulse coupled neural network (PCNN) special issue. *IEEE Trans Neural Netw* 10(3):461–463
33. Wang D, Freeman WJ, Kozma R, Lozowski A, Minai A (2004) Guest editorial special issue on temporal coding for neural information processing. *IEEE Trans Neural Netw* 15(5):953–956
34. Ranganath HS, Kuntimad G, Johnson JL (1995) Pulse coupled neural networks for image processing. In: *IEEE proceedings of Southeastcon'95 visualize the future*, pp 37–43.
35. Kuntimad G, Ranganath HS (1999) Perfect image segmentation using pulse coupled neural networks. *IEEE Trans Neural Netw* 10(3):591–598
36. Stewart RD, Fermin I, Oppen M (2002) Region growing with pulse-coupled neural networks: an alternative to seeded region growing. *IEEE Trans Neural Netw* 13(6):1557–1562
37. Ma Y, Dai R, Li L (2002) Automated image segmentation using pulse coupled neural networks and image's entropy. *J China Inst Commun* 23(1):46–51
38. Berg H, Olsson R, Lindblad T, Chilo J (2008) Automatic design of pulse coupled neurons for image segmentation. *Neurocomputing* 71(2008):1980–1993
39. Lu Y, Miao J, Duan L, Qiao Y, Jia R (2008) A new approach to image segmentation based on simplified region growing PCNN. *Appl Math Comput* 205(2):807–814
40. Shi M, Jiang S, Wang H, Xu B (2009) A simplified pulse-coupled neural network for adaptive segmentation of fabric defects. *Mach Vis Appl* 20(2):131–138
41. Wei S, Qu H, Hou M (2011) Automatic image segmentation based on PCNN with adaptive threshold time constant. *Neurocomputing* 74(2011):1485–1491
42. Chen Y, Park S-K, Ma Y, Ala R (2011) A new automatic parameter setting method of a simplified PCNN for image segmentation. *IEEE Trans Neural Netw* 22(6):880–892
43. Ranganath HS, Bhatnagar A (2011) Image segmentation using two-layer pulse coupled neural network with inhibitory linking field. *GSTF J Comput* 1(2):29–34
44. Zhao R, Ma Y (2012) A region segmentation method for region-oriented image compression. *Neurocomputing* 85:45–52
45. Gao C, Zhou D, Guo Y (2013) Automatic iterative algorithm for image segmentation using a modified pulse-coupled neural network. *Neurocomputing* 119(2013):332–338
46. Gao C, Zhou D, Guo Y (2014) An iterative thresholding segmentation model using a modified pulse coupled neural network. *Neural Process Lett* 39(1):81–95
47. Zhou D, Gao C, Guo Y (2014) A coarse-to-fine strategy for iterative segmentation using simplified pulse-coupled neural network. *Soft Comput* 18(3):557–570
48. Zhan K, Shi J, Li Q, Teng J, Wang M (2015) Image segmentation using fast linking SCM. In: *IEEE proceedings of IJCNN*, pp 2093–2100
49. Zhou D, Zhou H, Gao C, Guo Y (2015) Simplified parameters model of PCNN and its application to image segmentation. *Pattern Anal Appl*. doi:10.1007/s10044-015-0462-6
50. Helmy AK, El-Taweel GS (2016) Image segmentation scheme based on SOM-CNN in frequency domain. *Appl Soft Comput* 40:405–415
51. Ali JMH, Hassanien AE (2006) PCNN for detection of masses in digital mammogram. *Neural Netw World* 16(2):129
52. Murugavel M, Sullivan JM (2009) Automatic cropping of MRI rat brain volumes using pulse coupled neural networks. *Neuroimage* 45(3):845–854
53. Fu JC, Chen CC, Chai JW, Wong STC, Li IC (2010) Image segmentation by EM-based adaptive pulse coupled neural networks in brain magnetic resonance imaging. *Comput Med Imaging Graph* 34(4):308–320
54. Chou N, Wu J, Bai B, Qiu A, Chuang K-H (2011) Robust automatic rodent brain extraction using 3-D pulse-coupled neural networks (PCNN). *IEEE Trans Image Process* 20(9):2554–2564
55. Hage IS, Hamade RF (2013) Segmentation of histology slides of cortical bone using pulse coupled neural networks optimized by particle-swarm optimization. *Comput Med Imaging Graph* 37(7):466–474
56. Li J, Liu X, Zhuo J, Gullapalli RP, Zara JM (2013) An automatic rat brain extraction method based on a deformable surface model. *J Neurosci Methods* 218(1):72–82
57. Imamoglu N, Gomez-Tames J, Gonzalez J, Gu D, Yu W (2014) Pulse-coupled neural network segmentation and bottom-up saliency-on feature extraction for thigh magnetic resonance imaging based 3D model construction. *J Med Imaging Health Inform* 4(2):220–229
58. Harris MA, Van AN, Malik BH, Jabbour JM, Maitland KC (2015) A pulse coupled neural network segmentation algorithm for reflectance confocal images of epithelial tissue. *PloS One* 10(3):e0122368
59. Guo Y, Dong M, Yang Z, Gao X, Wang K, Luo C, Ma Y, Zhang J (2016) A new method of detecting micro-calcification clusters in mammograms using contourlet transform and non-linking simplified PCNN. *Comput Methods Programs Biomed* 130:31–45
60. Xie W, Li Y, Ma Y (2016) PCNN-based level set method of automatic mammographic image segmentation. *Optik* 127(4):1644–1650



61. Ranganath HS, Kuntimad G (1999) Object detection using pulse coupled neural networks. *IEEE Trans Neural Netw* 10(3):615–620
62. Yu B, Zhang L (2004) Pulse-coupled neural networks for contour and motion matchings. *IEEE Trans Neural Netw* 15(5):1186–1201
63. Ekblad U, Kinser JM, Atmer J, Zetterlund N (2004) The intersecting cortical model in image processing. *Nuclear Instrum Methods Phys Res Sect A* 525(1):392–396
64. Ekblad U, Kinser JM (2004) Theoretical foundation of the intersecting cortical model and its use for change detection of aircraft, cars, and nuclear explosion tests. *Signal Process* 84(7):1131–1146
65. Ji L, Zhang Y (2008) Fingerprint orientation field estimation using ridge projection. *Pattern Recognit* 41(5):1491–1503
66. Hassanien AE, Abraham A, Grosan C (2009) Spiking neural network and wavelets for hiding iris data in digital images. *Soft Comput* 13(4):401–416
67. Zhang X, Minai AA (2004) Temporally sequenced intelligent block-matching and motion-segmentation using locally coupled networks. *IEEE Trans Neural Netw* 15(5):1202–1214
68. Li Z, Hayward R, Zhang J, Liu Y, Walker R (2009) Towards automatic tree crown detection and delineation in spectral feature space using PCNN and morphological reconstruction. *IEEE Proc ICIP* 16:1705–1708
69. Hassanien AE, Al-Qaheri H, El-Dahshan E-SA (2011) Prostate boundary detection in ultrasound images using biologically-inspired spiking neural network. *Appl Soft Comput* 11(2):2035–2041
70. Ge W, Zhao H, Li X (2011) Gyroscope pivot bearing dimension and surface defect detection. *Sensors* 11(3):3227–3248
71. He D, Liu S, Liang X, Cai C (2011) Improved saliency toolbox/itti model for region of interest extraction. *Opt Eng* 50(9):097202–097202
72. Zhuang H, Low K-S, Yau W-Y (2012) Multichannel pulse-coupled-neural-network-based color image segmentation for object detection. *IEEE Trans Ind Electron* 59(8):3299–3308
73. Liu S, He D, Liang X (2012) An improved hybrid model for automatic salient region detection. *IEEE Signal Process Lett* 19(4):207–210
74. Gu X, Fang Y, Wang Y (2013) Attention selection using global topological properties based on pulse coupled neural network. *Comput Vis Image Underst* 117(10):1400–1411
75. Ni Q, Gu X (2014) Video attention saliency mapping using pulse coupled neural network and optical flow. In: *IEEEproceedings of IJCNN*, pp 340–344
76. Chen Y, Ma Y, Kim DH, Park S-K (2015) Region-based object recognition by color segmentation using a simplified PCNN. *IEEE Trans Neural Netw Learn Syst* 26(8):1682–1697
77. Karvonen JA (2004) Baltic sea ice SAR segmentation and classification using modified pulse-coupled neural networks. *IEEE Trans Geosci Remote Sens* 42(7):1566–1574
78. Li Z, Hayward R, Walker R, Liu Y (2011) A biologically inspired object spectral-texture descriptor and its application to vegetation classification in power-line corridors. *IEEE Geosci Remote Sens Lett* 8(4):631–635
79. Pratola C, Del Frate F, Schiavon G, Solimini D (2013) Toward fully automatic detection of changes in suburban areas from VHR SAR images by combining multiple neural-network models. *IEEE Trans Geosci Remote Sens* 51(4):2055–2066
80. Taravat A, Latini D, Del Frate F (2014) Fully automatic dark-spot detection from SAR imagery with the combination of nonadaptive weibull multiplicative model and pulse-coupled neural networks. *IEEE Trans Geosci Remote Sens* 52(5):2427–2435
81. Zhong Y, Liu W, Zhao J, Zhang L (2015) Change detection based on pulse-coupled neural networks and the NMI feature for high spatial resolution remote sensing imagery. *IEEE Geosci Remote Sens Lett* 12(3):537–541
82. Schäfer M, Schöner T, Wolff C, Hartmann G, Klar H, Rückert U (2002) Simulation of spiking neural networks-architectures and implementations. *Neurocomputing* 48(1):647–679
83. Schoenauer T, Atasoy S, Mehrtash N, Klar H (2002) Neuropipechip: a digital neuro-processor for spiking neural networks. *IEEE Trans Neural Netw* 13(1):205–213
84. Mehrtash N, Jung D, Hellmich HH, Schoenauer T, Lu VT, Klar H (2003) Synaptic plasticity in spiking neural networks (SP<sup>2</sup>INN): a system approach. *IEEE Trans Neural Netw* 14(5):980–992
85. Mehrtash N, Jung D, Klar H (2003) Image preprocessing with dynamic synapses. *Neural Comput Appl* 12(1):33–41
86. von der Malsburg C (1999) The what and why of binding: the modeler's perspective. *Neuron* 24(1):95–104
87. Chen L (2001) Perceptual organization: to reverse back the inverted (upside-down) question of feature binding. *Vis Cogn* 8(3–5):287–303
88. Elliffe MCM, Rolls ET, Stringer SM (2002) Invariant recognition of feature combinations in the visual system. *Biol Cybern* 86(1):59–71
89. Zhang J, Zhan K, Ma Y (2007) Rotation and scale invariant antinoise PCNN features for content-based image retrieval. *Neural Netw World* 2(07):121–132
90. Zhan K, Zhang H, Ma Y (2009) New spiking cortical model for invariant texture retrieval and image processing. *IEEE Trans Neural Netw* 20(12):1980–1986
91. Ma Y, Liu L, Zhan K, Wu Y (2010) Pulse-coupled neural networks and one-class support vector machines for geometry invariant texture retrieval. *Image Vis Comput* 28(11):1524–1529
92. Li X, Ma Y, Wang Z, Yu W (2012) Geometry-invariant texture retrieval using a dual-output pulse-coupled neural network. *Neural Comput* 24(1):194–216
93. Zhan K, Teng J, Ma Y (2013) Spiking cortical model for rotation and scale invariant texture retrieval. *J Inf Hiding Multimed Signal Process* 4(3):155–165
94. Gu X (2008) Feature extraction using unit-linking pulse coupled neural network and its applications. *Neural Process Lett* 27(1):25–41
95. Ebied HM, Revett K, Tolba MF (2013) Evaluation of unsupervised feature extraction neural networks for face recognition. *Neural Comput Appl* 22(6):1211–1222
96. Wang W, Zhou W, Zhao X (2014) Airplane extraction and identification by improved PCNN with wavelet transform and modified Zernike moments. *Imaging Sci J* 62(1):27–34
97. Mohammed MM, Badr A, Abdelhalim MB (2015) Image classification and retrieval using optimized pulse-coupled neural network. *Expert Syst Appl* 42(11):4927–4936
98. Srinivasan R, Kinser JM (1998) A foveating-fuzzy scoring target recognition system. *Pattern Recognit* 31(8):1149–1158
99. Allen FT, Kinser JM, Caulfield HJ (1999) A neural bridge from syntactic to statistical pattern recognition. *Neural Netw* 12(3):519–526
100. Rughooputh HCS, Rughooputh SDDV (2000) Spectral recognition using a modified Eckhorn neural network model. *Image Vis Comput* 18(14):1101–1103
101. Mureşan RC (2003) Pattern recognition using pulse-coupled neural networks and discrete Fourier transforms. *Neurocomputing* 51:487–493
102. Ursino M, Magosso E, Cuppini C (2009) Recognition of abstract objects via neural oscillators: interaction among topological

- organization, associative memory and gamma band synchronization. *IEEE Trans Neural Netw* 20(2):316–335
103. Wang X, Lei L, Wang M (2012) Palmprint verification based on 2D-Gabor wavelet and pulse-coupled neural network. *Knowl Based Syst* 27:451–455
104. Elons AS, Abull-Ela M, Tolba MF (2013) A proposed PCNN features quality optimization technique for pose-invariant 3D arabic sign language recognition. *Appl Soft Comput* 13(4):1646–1660
105. Tolba MF, Samir A, Aboul-Ela M (2013) Arabic sign language continuous sentences recognition using PCNN and graph matching. *Neural Comput Appl* 23(3–4):999–1010
106. Hou Y, Rao N, Lun X, Liu F (2014) Gait object extraction and recognition in dynamic and complex scene using pulse coupled neural network and feature fusion. *J Med Imaging Health Inform* 4(3):325–330
107. Wang Z, Sun X, Zhang Y, Zhu Y, Ma Y (2016) Leaf recognition based on PCNN. *Neural Comput Appl* 27(4):899–908
108. Li H, Jin X, Yang N, Yang Z (2015) The recognition of landed aircrafts based on PCNN model and affine moment invariants. *Pattern Recognit Lett* 51:23–29
109. Zhan K, Teng J, Shi J, Li Q, Wang M (2016) Feature-linking model for image enhancement. *Neural Comput* 28(6):1072–1100
110. Chacon MIM, Zimmerman AS (2003) Image processing using the PCNN time matrix as a selective filter. *IEEE Proc ICIP* 1:877–880
111. Gu X, Wang H, Yu D (2001) Binary image restoration using pulse coupled neural network. *Proc Neural Inf Process* 8:922–927
112. Ma Y, Shi F, Li L (2003) Gaussian noise filter based on PCNN. *IEEE Proc Neural Netw Signal Process* 1:149–151
113. Ma Y, Shi F, Li L (2003) A new kind of impulse noise filter based on PCNN. *IEEE Proc Neural Netw Signal Process* 1:152–155
114. Zhang J, Dong J, Shi M (2005) An adaptive method for image filtering with pulse-coupled neural networks. *IEEE Proc ICIP* 2:133–136
115. Ji L, Zhang Y, Shang L (2007) An improved pulse coupled neural network for image processing. *Neural Comput Appl* 17(3):255–263
116. Ji L, Zhang Y (2008) A mixed noise image filtering method using weighted-linking PCNNs. *Neurocomputing* 71(13):2986–3000
117. Zhang D, Nishimura TH (2010) Pulse coupled neural network based anisotropic diffusion method for 1/f noise reduction. *Math Comput Model* 52(11):2085–2096
118. Sang Y, Zhang Y, Zhou J (2010) Spatial point-data reduction using pulse coupled neural network. *Neural Process Lett* 32(1):11–29
119. Zhang D, Mabu S, Hirasawa K (2011) Image denoising using pulse coupled neural network with an adaptive Pareto genetic algorithm. *IEEJ Trans Electr Electron Eng* 6(5):474–482
120. Yuan J, Zhang H, Ma Y (2012) Effectual switching filter for removing impulse noise using a SCM detector. *Opt Eng* 51(3):037003
121. Padgett ML, Johnson JL (1997) Pulse coupled neural networks (PCNN) and wavelets: biosensor applications. *IEEE Proc ICNN* 4:2507–2512
122. Johnson JL, Padgett ML, Friday WA (1997) Multiscale image factorization. *IEEE Proc ICNN* 3:1465–1468
123. Johnson JL, Taylor JR, Anderson M (1999) Pulse-coupled neural network shadow compensation. In: *Proceedings of AeroSense*, International Society for Optics and Photonics pp 452–456
124. Gu X, Yu D, Zhang L (2005) Image shadow removal using pulse coupled neural network. *IEEE Trans Neural Netw* 16(3):692–698
125. Lindblad T, Kinser JM (1999) Inherent features of wavelets and pulse coupled networks. *IEEE Trans Neural Netw* 10(3):607–614
126. Broussard RP, Rogers SK (1996) Physiologically motivated image fusion using pulse-coupled neural networks. In: *Proceedings of SPIE, aerospace/defense sensing and controls*, International Society for Optics and Photonics, pp 372–383
127. Kinser JM (1997) Pulse-coupled image fusion. *Opt Eng* 36(3):737–742
128. Inguva R, Johnson JL, Schamschula MP (1999) Multifeature fusion using pulse-coupled neural networks. In: *AeroSense'99*, International Society for Optics and Photonics, pp 342–350
129. Broussard RP, Rogers SK, Oxley ME, Tarr GL (1999) Physiologically motivated image fusion for object detection using a pulse coupled neural network. *IEEE Trans Neural Netw* 10(3):554–563
130. Kinser JM (1999) Spiral image fusion by interchannel auto-waves. In: *Ninth workshop on virtual intelligence/dynamic neural networks: neural networks fuzzy systems, evolutionary systems and virtual Re*, International Society for Optics and Photonics, vol 9, pp 148–154
131. Li M, Cai W, Tan Z (2006) A region-based multi-sensor image fusion scheme using pulse-coupled neural network. *Pattern Recognit Lett* 27(16):1948–1956
132. Huang W, Jing Z (2007) Multi-focus image fusion using pulse coupled neural network. *Pattern Recognit Lett* 28(9):1123–1132
133. Yang S, Wang M, Lu Y, Qi W, Jiao L (2009) Fusion of multiparametric SAR images based on SW-nonsubsampled contourlet and PCNN. *Signal Process* 89(12):2596–2608
134. Agrawal D, Singhai J (2010) Multifocus image fusion using modified pulse coupled neural network for improved image quality. *IET Image Process* 4(6):443–451
135. Chang W, Guo L, Fu Z, Liu K (2010) Hyperspectral multi-band image fusion algorithm by using pulse coupled neural networks. *J Infrared Millim Waves* 29(3):205–209,235
136. Yang S, Wang M, Jiao L, Wu R, Wang Z (2010) Image fusion based on a new contourlet packet. *Inf Fusion* 11(2):78–84
137. Chai Y, Li HF, Qu JF (2010) Image fusion scheme using a novel dual-channel PCNN in lifting stationary wavelet domain. *Opt Commun* 283(19):3591–3602
138. Chai Y, Li HF, Guo MY (2011) Multifocus image fusion scheme based on features of multiscale products and PCNN in lifting stationary wavelet domain. *Opt Commun* 284(5):1146–1158
139. Yang S, Wang M, Jiao L (2012) Contourlet hidden Markov tree and clarity-saliency driven PCNN based remote sensing images fusion. *Appl Soft Comput* 12(1):228–237
140. Geng P, Wang Z, Zhang Z, Xiao Z (2012) Image fusion by pulse couple neural network with shearlet. *Opt Eng* 51(6):067005
141. Das S, Kundu MK (2012) NSCT-based multimodal medical image fusion using pulse-coupled neural network and modified spatial frequency. *Med Biol Eng Comput* 50(10):1105–1114
142. Das S, Kundu MK (2013) A neuro-fuzzy approach for medical image fusion. *IEEE Trans Biomed Eng* 60(12):3347–3353
143. El-taweel GS, Helmy AK (2013) Image fusion scheme based on modified dual pulse coupled neural network. *IET Image Process* 7(5):407–414
144. Kang B, Zhu W, Yan J (2013) Fusion framework for multi-focus images based on compressed sensing. *IET Image Process* 7(4):290–299
145. Lin Z, Yan J, Yuan Y (2013) Algorithm for image fusion based on orthogonal grouplet transform and pulse-coupled neural network. *J Electron Imaging* 22(3):033028

146. Shi C, Miao Q, Xu P (2013) A novel algorithm of remote sensing image fusion based on shearlets and PCNN. *Neurocomputing* 117:47–53
147. Kong W, Liu J (2013) Technique for image fusion based on nonsubsampling shearlet transform and improved pulse-coupled neural network. *Opt Eng* 52(1):017001
148. Zhang B, Zhang C, Liu Y, Wu J, He L (2014) Multi-focus image fusion algorithm based on compound pcnn in surfacelet domain. *Optik* 125(1):296–300
149. Zhang B, Lu X, Jia W (2013) A multi-focus image fusion algorithm based on an improved dual-channel PCNN in NSCT domain. *Optik* 124(20):4104–4109
150. Zhang B, Zhang C, Wu J, Liu H (2014) A medical image fusion method based on energy classification of BEMD components. *Optik* 125(1):146–153
151. Zhao Y, Zhao Q, Hao A (2014) Multimodal medical image fusion using improved multi-channel PCNN. *Biomed Mater Eng* 24(1):221–228
152. Kong W, Zhang L, Lei Y (2014) Novel fusion method for visible light and infrared images based on NSST-SF-PCNN. *Infrared Phys Technol* 65:103–112
153. Lang J, Hao Z (2014) Novel image fusion method based on adaptive pulse coupled neural network and discrete multi-parameter fractional random transform. *Opt Lasers Eng* 52:91–98
154. Zhang X, Li X, Feng Y, Zhao H, Liu Z (2014) Image fusion with internal generative mechanism. *Expert Syst Appl* 42(5):2382–2391
155. Yin H, Liu Z, Fang B, Li Y (2015) A novel image fusion approach based on compressive sensing. *Opt Commun* 354:299–313
156. Ganasala P, Kumar V (2016) Feature-motivated simplified adaptive PCNN-based medical image fusion algorithm in NSST domain. *J Digit Imaging* 29(1):73–85
157. Lang J, Hao Z (2015) Image fusion method based on adaptive pulse coupled neural network in the discrete fractional random transform domain. *Optik* 126(23):3644–3651
158. Peng G, Wang Z, Liu S, Zhuang S (2015) Image fusion by combining multiwavelet with nonsubsampling direction filter bank. *Soft Comput*. doi:[10.1007/s00500-015-1893-0](https://doi.org/10.1007/s00500-015-1893-0)
159. Koch C, Segev I (2000) The role of single neurons in information processing. *Nature Neurosci* 3:1171–1177
160. Gove A, Grossberg S, Mingolla E (1995) Brightness perception, illusory contours, and corticogeniculate feedback. *Vis Neurosci* 12(06):1027–1052
161. Barnes T, Mingolla E (2013) A neural model of visual figure-ground segregation from kinetic occlusion. *Neural Netw* 37:141–164
162. Brosch T, Neumann H (2014) Interaction of feedforward and feedback streams in visual cortex in a firing-rate model of columnar computations. *Neural Netw* 54:11–16
163. French AS, Stein RB (1970) A flexible neural analog using integrated circuits. *IEEE Trans Biomed Eng* 3:248–253
164. Kinser JM (1996) A simplified pulse-coupled neural network. *Proc SPIE* 2760:563–567
165. Gu X, Yu D, Zhang L (2004) Image thinning using pulse coupled neural network. *Pattern Recognit Lett* 25(9):1075–1084
166. Ji L, Zhang Y, Shang L, Pu X (2007) Binary fingerprint image thinning using template-based PCNNs. *IEEE Trans Syst Man Cybern Part B Cybern* 37(5):1407–1413
167. Shang L, Zhang Y, Ji L (2007) Binary image thinning using autowaves generated by PCNN. *Neural Process Lett* 25(1):49–62
168. Shang L, Zhang Y, Ji L (2009) Constrained ZIP code segmentation by a PCNN-based thinning algorithm. *Neurocomputing* 72(7):1755–1762
169. Caulfield HJ, Kinser JM (1998) Finding the shortest path in the shortest time using PCNN's. *IEEE Trans Neural Netw* 10(3):604–606
170. Qu H, Yang SX, Willms AR, Zhang Y (2009) Real-time robot path planning based on a modified pulse-coupled neural network model. *IEEE Trans Neural Netw* 20(11):1724–1739
171. Zhang J, Zhao X, He X (2014) A minimum resource neural network framework for solving multiconstraint shortest path problems. *IEEE Trans Neural Netw Learn Syst* 25(8):1566–1582
172. McEniry R, Johnson JL (1997) Methods for image segmentation using a pulse coupled neural network. *Neural Netw World* 2(97):177–189
173. Wang D (2005) The time dimension for scene analysis. *IEEE Trans Neural Netw* 16(6):1401–1426
174. Rybak IA, Shevtsova NA, Podladchikova LN, Golovan AV (1991) A visual cortex domain model and its use for visual information processing. *Neural Netw* 4(1):3–13
175. Rybak IA, Shevtsova NA, Sandler VM (1992) The model of a neural network visual preprocessor. *Neurocomputing* 4(1–2):93–102
176. Wang D, Terman D (1997) Image segmentation based on oscillatory correlation. *Neural Comput* 9(4):805–836
177. Brodatz P (1966) Textures: a photographic album for artists and designers. Dover Publications, New York
178. Gonzalez RC, Woods RE, Eddins SL (2009) Digital image processing using MATLAB, 2nd edn. Gatesmark Publishing, New Jersey
179. Ma Y, Lin D, Zhang B, Liu Q, Gu J (2007) A novel algorithm of image gaussian noise filtering based on PCNN time matrix. In: *IEEE proceedings of signal processing and communications*, pp 1499–1502
180. Marr D (1982) Vision: a computational investigation into the human representation and processing of visual information. Henry H and Co., Inc, New York