Color Image Segmentation Using Pulse-Coupled Neural Network for Locusts Detection

Xuemei Xiong and Yiming Wang
College of Information and Electrical Engineering
China Agricultural University
Beijing, China

Xiaochao Zhang
Institute of Mechanical & Electrical Technology
Academy of Agricultural Mechanization Sciences
Beijing, China

Abstract - *Pulse-coupled neural network (PCNN) is different from traditional artificial neural networks, which can be applied in many fields, such as image processing. A crucial step in developing automated locust detection system is image segmentation, which aims to identify locusts from images. In this study, a simplified PCNN algorithm was proposed for this purpose. Color images are either converted to gray-scale images or to the hue, saturation, intensity (HSI) color space before being segmented by the PCNN algorithm. To measure the quality of segmentation, the area recognition rate (ARR) criterion was computed. The ARR achieved 94 percent on the locust image*[1].

Keywords: Pulse-coupled neural network, Image Segmentation.

1 Introduction

With the development of new agricultural areas in China, many local locusts acquire the status of pests of cultivated plants. Their cumulative feeding effect can cause significant damage to the crop, decorative trees, desert shrubs and other plants. Most damage is caused by young larvae in nurseries. The need for survey and control is stressed. Early detection is an important step to control the disaster. For this purpose, an automated locust detection system which uses aerocraft that hedgehops above the wetland where investigators cannot arrive at is under establishment. The framework of the system includes two primary components: image data acquisition and image data processing. Image segmentation is often the most time-consuming part of the process. It is necessary to develop new segmentation algorithms that can separate all locusts from images. Pulse-coupled neural network (PCNN) is a biologically inspired neural net. It is different from what we generally mean by artificial neural networks in the sense that it does not train. It has similarities to the biological vision system, which extracts various features in parallel by different centra and then fuses them at the stage of consciousness. 2-D images can be transformed into 1-D time signals with some interesting features. The goal for locust detection is to eventually identify objects as in images. It is generally far easier to be accomplished by examining the pulse outputs of the PCNN rather than the original image. Thus, the PCNN becomes a very useful tool. Most PCNNs for image segmentation have been developed for gray-scaled images. Color image segmentation algorithms are based on gray level (monochrome) segmentation approaches, which can be directly applied to each component of a color space [1]. The selection of a color space is application dependent. In this work, we propose a color image segmentation method based on PCNN. The area recognition rate (ARR) is used to measure the performance of this method.

2 Image processing using PCNN

In the field of image processing, traditional models are either subject to problems determined by geometric transforms (scaling, translation or rotation) or to high computational complexity. PCNN as a neural model adopts parallel processing that can solve the problem of computational complexity. It is quite feasible to implement it in specialized hardware due to its local connections [2] [3][4] [5] [6].

PCNN can be used in several tasks in the field of image processing, such as target recognition, segmentation, edge extraction, object identification, object isolation and locating foveation points in images. The use of PCNN for identification of objects was introduced by J. L. Johnson [7] and the work has been preceded by Jason Kinser and Thomas Lindblad [8]. In [9] a method for image segmentation that combines the PCNN with between-cluster variance can achieve better image segmentation not only for large objects (Lena image) but also for small objects (Tank IR image). Therefore it can be applied universally. In [10] a model consists of a PCNN, a discrete Fourier transform module and a multilayer perceptron classifier is used for pattern recognition. The system showed total translation independence.

2.1 PCNN model

When PCNN is applied in image processing, it is a single layer two-dimensional array of laterally linked

[1] This work was supported in part by the Ministry of Science and Technology of the People´s Republic of China under Grant 2004DIB3J076

neurons. The number of neurons in the network is equal to the number of pixels in the input image. One-to-one correspondence exists between image pixels and neurons. Each pixel is connected to a unique neuron and each neuron is connected with the surrounding neurons with a radius of linking field (Fig.1). The neuron receives input signals from other neurons and from external sources through the receptive fields. After the receptive fields have collected the inputs, they are divided into two or more internal channels. One channel is the feeding input *F* and the other is the linking input *L*. The distinction between the feeding and the linking is that the feeding connections are required to have a slower characteristic response time constant than those of the linking inputs. The linking inputs are bia sed and then multiplied together, and further multiplied with the feeding input to form the total internal activity *U*. The pulse generator of the neuron consists of a step-function generator and a threshold signal generator. At each time step the neuron output *Y* is set to 1 when the internal activity *U* is greater than the threshold function *T*. The threshold input at each time step is updated. The output of the neuron is consequently reset to zero when *T* is larger than *U*. Thus at one time step the pulse generator produces a single pulse at its output whenever the value of *U* exceeds *T*.

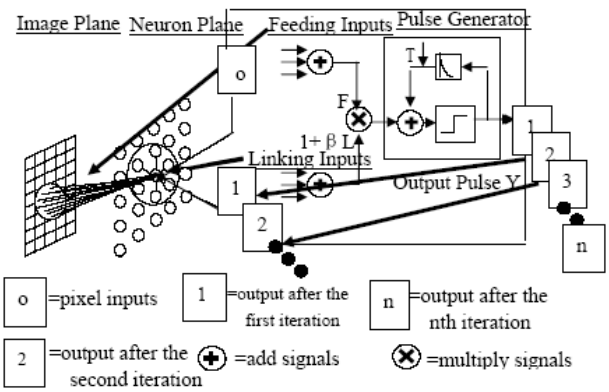

Fig.1 Structure of PCNN

The functionality is obtained simply by iterating over equations (1)–(5):

$$F_{ij}[n] = e^{-\alpha_F} F_{ij}[n-1] + S_{ij} + V_F \sum_{kl} M_{ijkl} Y_{kl}[n-1] \quad (1)$$

$$L_{ij}[n] = e^{-\alpha_L} L_{ij}[n-1] + V_L \sum_{kl} W_{ijkl} Y_{kl}[n-1] \quad (2)$$

$$U_{ij}[n] = F_{ij}[n]\left(1 + \beta L_{ij}[n]\right) \quad (3)$$

$$Y_{ij}[n] = \begin{cases} 1 & U_{ij}[n] > T_{ij}[n] \\ 0 & U_{ij}[n] \le T_{ij}[n] \end{cases} \quad (4)$$

$$T_{ij}[n] = e^{-\alpha_T} T_{ij}[n-1] + V_T Y_{ij}[n] \ . \quad (5)$$

We refer to *n* as being the current iteration where *n* varies from 1 to *N-1* (*N* is the total number of iterations; *n*=0 is the initial state). The (*i*,*j*) pare stands for the position of the neuron in the map. α_F , α_L and α_T are time constants. V_F , V_L and V_T are normalizing constants. *M* and *W* represent the constant synaptic weights.

Each iteration updates the internal activity and the output for every neuron in the network, based on the stimulus signal from the image and the previous state of the network. For each iteration the total number of firings over the entire PCNN is computed and stored in a global array *G*:

$$G[n] = \sum_{ij} Y_{ij}[n] \ . \quad (6)$$

2.2 Simplification of the parameters

Note, each neuron only fires once. The normalizing constants V_F and V_L effect the multi-fire frequency. Since we use the single-pass model of PCNN, which means that each pixel fires once, V_F and V_L can be left out of account.

Like in [11], we consider only the external input as feeding input. Therefore Eq.1 is replaced by:

$$F_{ij}[n] = S_{ij} \ . \quad (7)$$

Note that each neuron is connected with neurons by the L channel in its neighbor field. If one or more than one neuron fire in the neighbor field of neuron N_{ij} , let L=1;else, let L=0. Therefore, the linking input is updated by a step function:

$$L_{ij}[n] = Step(Y_{ij}[n-1] * W_{ijkl}) \ , \quad (8)$$

where * is two-dimensional convolution; *K* is a square matrix whose dimension is ($2(radius\ of\ linking\ field)+1$) with center value 1; other pixel values are *l/r* where *r* is the distance from the center pixel. The convolution between *Y* and *W* determines the local intensity of surrounding neurons firing at the same time. In Eq.(3); β is the linking strength, and β plays an important role in segmentation of images[12]. In each iteration, the linking strength β is updated by a factor α . Let $\beta = \alpha\beta$ and α is factor of linking decay. The exponential decay of threshold *T* is replaced by a linear decay:

$$T_{ij}[n] = T_{ij}[n-1] - \Delta T + V_T Y_{ij}[n-1] \ , \quad (9)$$

where ΔT is the constant for threshold decay.

The PCNN is simplified and the parameters of PCNN that need to be set are *W*, α , β , V_T and ΔT . We use the same initial *W* for these two image segmentation processing:

$$W = \begin{pmatrix} 0.707 & 1 & 0.707 \\ 1 & 1 & 1 \\ 0.707 & 1 & 0.707 \end{pmatrix}. \qquad (10)$$

The process above is implemented in Matlab 6.

3 Using PCNN to identify locusts

The pictures taken by aerocraft that hedgehops above the wetland contains small objects (Fig.2) and large objects (Fig.3) due to the distance between the altitude of the aerocraft and the resolution of the camera. The RGB color images are processed in two ways for the purpose of comparison and finding of best way to cheat different images:

1) Convert the image to a gray-scale image; and then segment the gray-scale image with the gray-PCNN algorithm (fig.2 and fig. 3).
2) Convert the image to HSI space; extract the individual component images; segment the H, S, I component images respectively by the color-PCNN algorithm (fig.4 and fig. 5).

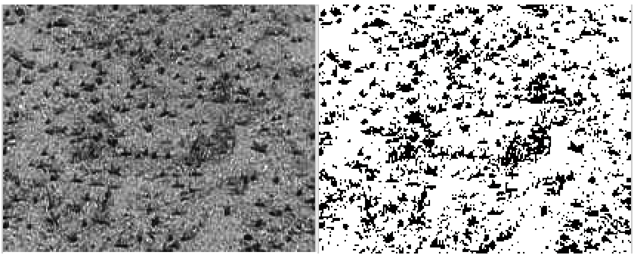

Fig.2 Locusts from a long distance(left, converted as a gray-scale image) and the segmentation result(right, *n*=2)

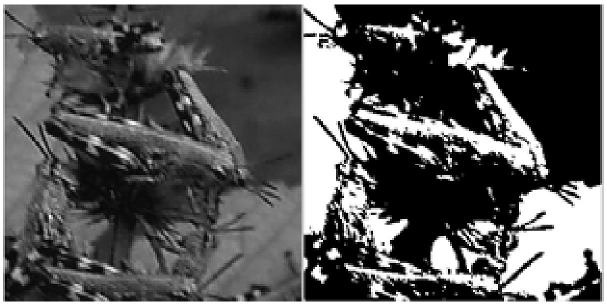

Fig.3 Locusts from a short distance (left, converted as a gray-scale image) and the segmentation result (right, *n*=2)

For tiny objects (fig.4), only the saturate channel was used in the process since it performed best at separating locusts from background. For large objects, the intensity component image by a color-PCNN algorithm (fig.5) has the same quality as the result by a gray-PCNN algorithm (fig.3).

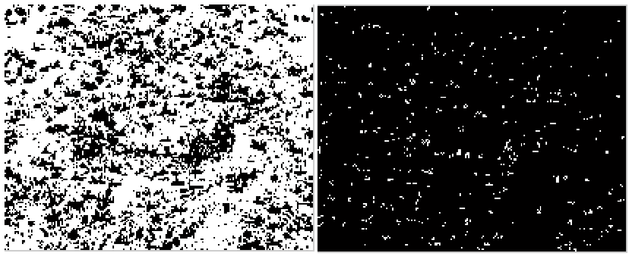

Fig. 4 Outputs of a selected PCNN iteration(*n*=2) of the extracted component images from HSI space (intensity image on left, saturation image on right)

Fig. 5 Outputs of a selected PCNN iteration of the extracted component images from HSI space (intensity image(*n*=2) on left, saturation image(*n*=4) on right)

Table 1 Performance of PCNN on images of locusts

Images in	Parameters	ARR	Average processing time
Fig2	$\Delta T = 0.02 \quad V_T = 100$ $\beta = 0.8, \alpha = 0.2$	92%	1 s
Fig3	$\Delta T = 0.02 \quad V_T = 100$ $\beta = 0.08, \alpha = 0.2$	81 %	1 s
Fig4	$\Delta T = 0.02 \quad V_T = 100$ $\beta = 0.8, \alpha = 0.2$	94 %	1 s
Fig5	$\Delta T = 0.02 \quad V_T = 100$ $\beta = 0.08, \alpha = 0.2$	81 %	1 s

In a binary image, in general, the bright region corresponds to the background and the dark region corresponds to the object, such as locusts or in reverse (saturation component image in fig.4). The aero recognition rate (ARR) is used to measure the segmentation quality. The ARR criterion is defined as:

$$ARR = (1 - \frac{wp}{tp}) \times 100\% \;, \qquad (11)$$

where *wp* is the number of incorrectly recognized pixels in the image and *tp* is the total number of pixels in the image.

The performance of PCNN on locust images is shown in Table 1. The best quality of image segmentation for tiny objects (fig.2,4) is obtained by a relatively larger linking strength β. The last column of Table 1 shows the average processing time per image. Our evaluation was performed on a 1.8GHz Pentium with 256 Mbytes of RAM ,Windows

Xp and Matlab 6. It generally took 0.5 s to produce a single binary image. One run through the whole image required 2 iterations to achieve the best binary image in the shown experiment. Thus the minimum total time to process a grayscale image of 256×256 pixels size is about 1s. Since color-PCNN image segmentation algorithm is based on gray level (monochrome) segmentation algorithm, the average processing time is about the same as the gray-PCNN algorithm. The average processing time could be reduced by using C programming instead of Matlab.

Figure 5 shows an image with shadow. The result of PCNN can be improved by utilizing a PCNN variation in [13]. Once the locusts have been located and identified and the density of locusts has been calculated, the spray process at the aerocraft has to follow the main detection system.

4 Conclusions

We proposed a simplified gray-PCNN algorithm and a color-PCNN algorithm for locusts images by simplifying the parameters of PCNN and introducing a factor for the linking strength. The performance of PCNN is tested with two images of locusts. The ARR achieved 94 percent on the locust image. The proper setting of the various parameters of the network, such as linking parameters, thresholds, and interconnection matrices, depending on the input image, can make PCNN image processing more efficient.

We have to mention that the main purpose of this paper is to show the possibility of detecting locusts presented in images using PCNN. Experiments show that PCNN can be used to solve problems related to locust detection. This paper also expands the detecting method of locust.

In general, the PCNN algorithm should be viewed as a 'preprocessor' that needs to be combined with other image processing transforms to be a complete system. Future research will be on the combination of PCNN with other image processing transforms to get better segmentation of locusts from background.

5 References

[1] Cheng H.D., Jiang X. H., Sun Y. and Wang J. Color image segmentation: Advances and prospects. Pattern Recognition, 34:2259-2281,2001.

[2] Frank C., Hartmann G., Jahnke A. and Schafer M. An accelerator for neural networks: With pulse-coupled model neurons. *IEEE Trans.Neural Networks*, 10(3):527-538,1999.

[3] Waldemark J., Millberg M., Lindblad T., Waldemark K. and Becanovic V. Implementation of a pulse coupled neural network in FPGA.*Int.J Neural Syst.*, ,10(3):171-177,2000.

[4] Ota Y. and Wilamowski B. M. Analog implementation of pulse-coupled neural networks . I*EEE Trans. Neural Networks*, 1999,10(3):539-543

[5] John J. L. and Ritter D. Observation of periodic waves in a Pulse-coupled neural network. *Opt. Lett.*,18（15）:1253-1255, 1993.

[6] Johnson J. L. Pulse-coupled neural networks, Adaptive Comput.: Math., Electron., Opt., SPIE. *Crit.Rev.*, (CR55): 47-76,1994.

[7] Johnson, J.L. Time Signatures of Images, Proceedings, IEEE International Conference on Neural Networks, Orlando, FL. 1994.

[8] Lindblad, Th. and Kinser, J. M. Image Processing using Pulse- Coupled Neural Networks, *Perspectives In Neural Computing*. Springer-Verlag Limited. ISBN 3-540-76264-7.

[9] Wang H., Zhang K. and Li Y. Image segmentation method based on PCNN. *Opto-Electronic Engineering*, , 32(5): 93-96,2005.

[10] Raul C. Muresan Pattern recognition using pulse-coupled neural networks and discrete Fourier transforms. *Neurocomputing*, 51 : 487 － 493,2003.

[11] Gu X., Yu D. and Zhang L. Image thinning using pulse coupled neural network. *Pattern Recognition Letters* ,25:1075-1084, 2004.

[12] Gu X., Wang H. and, Yu D. A New Approach for Image Shadow Processing Based on PCNN. *Journal of Electronics and Information Technology*, 26(3): 479-483, 2004.

[13] Chen J. , Gong J. and Liu X. Image Binary Segmentation by Simplified PCNN. *Journal of Geomatics*,30(1):38-40,2005.