Hopfield neural network and its applications on image edge detection

Yonghong Zhang (张永宏), Dejin Hu (胡德金), Kai Zhang (张 凯), and Junjie Xu (徐俊杰)

School of Mechanical and Power Engineering, Shanghai Jiaotong University, Shanghai 200030

Received October 13, 2003

A method of image edge detection using the Hopfield neural network (HNN) is proposed in this paper. The image edge parameters are introduced in detail, and the energy function of HNN is given based on the edge parameters. Tests on the image edge detection show that images detected by the proposed method have good edge closeness and true edge, at the same time it has good anti-noise performance. The image edge detection using HNN is better than that obtained by some other edge detection operators.

OCIS codes: 100.2000, 110.2970, 100.2960, 100.3010.

Edge detection is one of the most important research aspects in the area of image processing and computer vision, so it has attracted much attention. There are some perfect edge detection methods in theory, such as Marr operator and Canny operator[1,2]. These classic edge detection methods are used to construct detection operator in certain neighboring area of pixels. From results of some experiments we found that the edge is inaccurate for complex image, edge detail is easy to lose, further more, the noise in image has a biggish effect for the result of image edge detection. For above reasons, new methods for image edge detection are needed to improve survey availability. Researchers presented some detection methods along with studies on article neural network, such as methods using hybrid neural network, information measure, and BP neural network. In fact, there are some shortages for them. For example the BP operator convergence speed is slow and it is easy to converge local infinitesimal position. This method has low steady data and unregulated parameters. So their applications are limited.

In this paper, a new method of Hopfield neural network (HNN) and its application on image edge detection are presented. Parameters of image edge detection are decided at first. We give the energy function of HNN based on the edge parameters according to the principle of HNN. Finally, we can get the exact image edge availably after the neural network was studied and trained according to the stability of energy function.

The principle of image detection using HNN is that pixel gray levels are discontinuous in the area of image edge. According to the principle, a HNN energy function can be founded. Muneyyasu and Uchiyama presented edge detection parameters and founded energy function of Hopfield neural network[3,4]. Figures 1 and 2 show two and four edge detection parameters, respectively. After some experiments, we found that the image edge detection quality is not perfect using these two methods. The first one can detect horizontal and vertical edge details exactly, but it

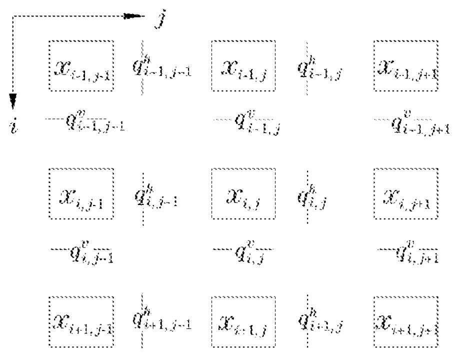

Fig. 1. Edge parameters q^h, q^v, and pixels $x_{i,j}$.

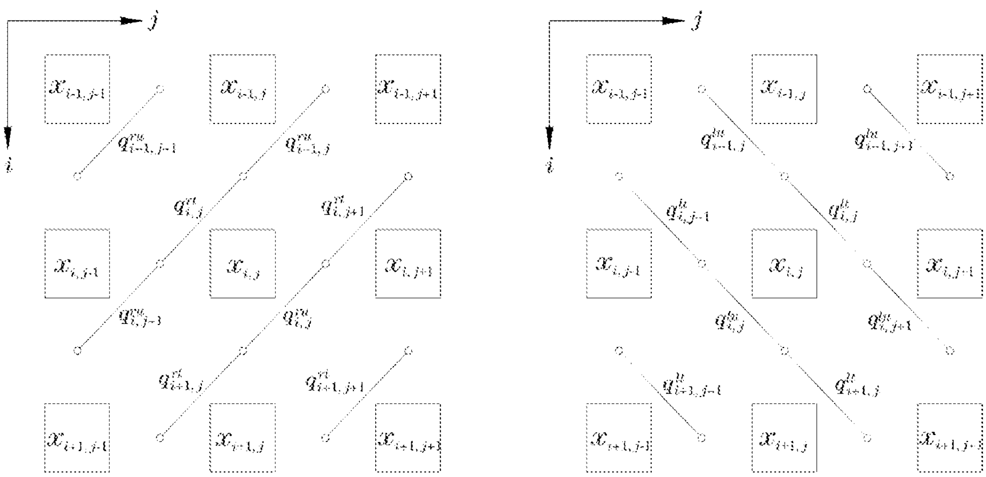

Fig. 2. Edge parameters $q^{ru}, q^{rt}, q^{lu}, q^{lt}$, and pixels $x_{i,j}$.

may lose slanting edge easily and its closeness is not perfect. The other methods can detect slanting edge details and has perfect anti-noise performance, while its disadvantage is easy to lose horizontal and vertical edge details.

In order to improve image edge detection qualities, the image edge details should be preserved perfectly. Noise should be restrained availably. According to these factors, we presented edge detection parameters q of two-dimensional (2D) gray image, here $q = [q^h, q^v, q^m, q^n, q^{ru}, q^{rt}, q^{lu}, q^{lt}]$. Figure 3 shows the parameters setting, $x_{i,j}$ is the pixel gray level where the pixel is located at i row j column. The edge detection parameters are introduced as follows.

1) The parameter q^h is used to detect horizontal image edge, q^v is used to detect vertical image edge. At the horizontal direction, if gray level difference of two conjoint pixels satisfies $|x_{i,j} - x_{i,j+1}| > T$, then $q^h_{i,j} = 1$, here T is threshold. At this time, we consider that there is an edge between pixels $x_{i,j}$ and $x_{i,j+1}$. Otherwise $q^h_{i,j} = 0$, that is to say, there is not any edge between pixels $x_{i,j}$ and $x_{i,j+1}$. The threshold T is determined by sharp degree of gray level variety between edges in images. At the vertical orientation, there are alike instances. If gray level difference of two conjoint pixels satisfies $|x_{i,j} - x_{i+1,j}| > T$, then $q^v_{i,j} = 1$, that is to say, there is an edge between pixel $x_{i,j}$ and $x_{i+1,j}$. Otherwise $q^v_{i,j} = 0$.

2) We suppose that ε is a minimal positive number. If $|x_{i,j+1} + x_{i,j-1} - 2x_{i,j}| < \varepsilon$ and $|x_{i,j+1} - x_{i,j-1}| > T$, it shows that the image edge lies on the pixel $x_{i,j}$. Here $q^m_{i,j} = 1$. Otherwise $q^m_{i,j} = 0$. On the other hand, if $|x_{i-1,j} + x_{i+1,j} - 2x_{i,j}| < \varepsilon$ and $|x_{i+1,j} - x_{i-1,j}| > T$, this means that the image edge also lies on the pixel $x_{i,j}$ and $q^n_{i,j} = 1$. Otherwise $q^n_{i,j} = 0$.

3) Parameters q^{ru}, q^{rt}, q^{lu}, q^{lt} are used to detect slanting direction edge details in the surrounding area of pixel respectively. If the gray levels of conjoint pixels $x_{i,j}$, $x_{i,j+1}$, and $x_{i+1,j}$ meet requirement of $|x_{i,j} - x_{i,j+1}| > T$ and $|x_{i,j} - x_{i+1,j}| > T$, then $q^{ru}_{i,j} = 1$, we call that there is an edge at the right lower direction of pixel $x_{i,j}$, otherwise $q^{ru}_{i,j} = 0$ and there is not any edge. The parameters q^{rt}, q^{lu}, q^{lt} are also controlled by the same manner as q^{ru} is.

We found that each direction details are all considered through the above parameters analysis. So we can get integrated image edge by this method.

As stated above, the edge parameters are decided by minimizing the energy function of an HNN. According to the edge detection parameters, we found the following energy functions of HNN:

$$E = \sum E^i \quad (i = h, v, m, n, ru, rt, lu, lt). \tag{1}$$

Here

$$\begin{cases} E^h = E^h_s + E^h_p \\ E^v = E^v_s + E^v_p \end{cases}, \tag{2}$$

$$\begin{cases} E^m = E^m_s + E^m_p \\ E^n = E^n_s + E^n_p \end{cases}, \tag{3}$$

$$\begin{cases} E^{ru} = E^{ru}_s + E^{ru}_p \\ E^{rt} = E^{rt}_s + E^{rt}_p \end{cases}, \tag{4}$$

$$\begin{cases} E^{lu} = E^{lu}_s + E^{lu}_p \\ E^{lt} = E^{lt}_s + E^{lt}_p \end{cases}, \tag{5}$$

where E^h and E^v are energy functions as to horizontal direction edge parameter q^h and vertical direction edge parameter q^v respectively, and E^m and E^n are energy functions as to edge parameters q^m and q^n, here the image edge lies on the pixel exactly. E^{ru}, E^{rt}, E^{lu}, and E^{lt} are energy functions to decide the slanting direction edge parameters q^{ru}, q^{rt}, q^{lu} and q^{lt}, respectively.

According to the relations between edge parameters and pixel gray levels, the variable E^i ($i = h, v, m, n, ru, rt, lu, lt$) appearing in Eqs. (2)$-$(5) are defined as follows.

E^h_s, E^m_s: the functions to decide horizontal edge parameters. They are defined as

$$E^h_s = \sum_i \sum_j [(1 - q^h_{i,j})(x_{i,j} - x_{i,j+1})^2 + (1 - q^h_{i,j-1})(x_{i,j} - x_{i,j-1})^2], \tag{6}$$

$$E^m_s = \sum_i \sum_j [(1 - q^m_{i,j})(x_{i,j-1} + x_{i,j+1} - 2x_{i,j})^2]. \tag{7}$$

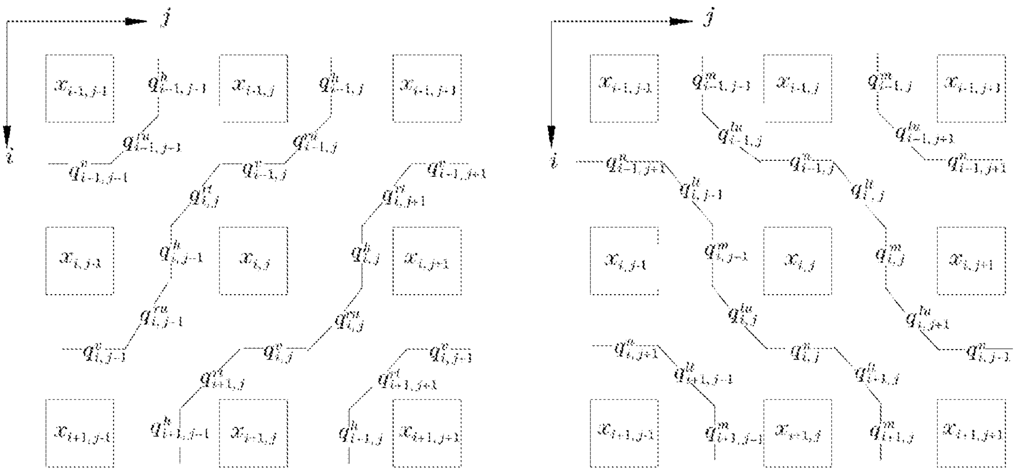

Fig. 3. Edge parameters q^h, q^v, q^m, q^n, q^{ru}, q^{rt}, q^{lu}, q^{lt}, and pixels $x_{i,j}$.

E_s^v, E_s^n: the functions to decide vertical edge parameters. They are defined as

$$E_s^v = \sum_i \sum_j [(1 - q_{i,j}^v)(x_{i,j} - x_{i+1,j})^2 + (1 - q_{i-1,j}^h)(x_{i,j} - x_{i-1,j})^2], \quad (8)$$

$$E_s^n = \sum_i \sum_j [(1 - q_{i,j}^n)(x_{i-1,j} + x_{i+1,j} - 2x_{i,j})^2]. \quad (9)$$

E_p^h, E_p^m: the functions to restrict horizontal edge location and total number of the edges. They are defined as

$$\begin{aligned} E_p^h = {} & C_p \sum_i \sum_j (q_{i,j}^h q_{i,j+1}^h + q_{i,j}^h q_{i,j-1}^h + q_{i,j-1}^h q_{i,j-2}^h) \\ & + C_c \sum_i \sum_j (q_{i,j}^h + q_{i.j-1}^h) \\ & + \frac{1}{2} C_l \sum_i \sum_j \{ q_{i,j}^h [(1 - q_{i+1,j}^h - q_{i,j}^v - q_{i,j+1}^v)^2 \\ & + (1 - q_{i-1,j}^h - q_{i-1,j}^v - q_{i-1,j+1}^v)^2] \\ & + q_{i,j-1}^h [(1 - q_{i+1,j-1}^h - q_{i,j-1}^v - q_{i,j}^v)^2 \\ & + (1 - q_{i-1,j-1}^h - q_{i-1,j-1}^v - q_{i-1,j}^v)^2] \}, \end{aligned} \quad (10)$$

$$\begin{aligned} E_p^m = {} & C_p \sum_i \sum_j (q_{i,j}^m q_{i,j+1}^m + q_{i,j}^m q_{i,j-1}^m) + C_c \sum_i \sum_j q_{i,j}^m \\ & + C_l \sum_i \sum_j q_{i,j}^m [(1 - q_{i,j+1}^m - q_{i,j-1}^m)^2 \\ & + (1 - q_{i+1,j}^m - q_{i-1,j}^m)^2]. \end{aligned} \quad (11)$$

E_p^v, E_p^n: the functions to restrict vertical edge location and total number of the edges. They are defined as

$$\begin{aligned} E_p^v = {} & C_p \sum_i \sum_j (q_{i,j}^v q_{i+1,j}^v + q_{i,j}^v q_{i-1,j}^v + q_{i-1,j}^v q_{i-2,j}^v) \\ & + C_c \sum_i \sum_j (q_{i,j}^v + q_{i-1.j}^v) \\ & + \frac{1}{2} C_l \sum_i \sum_j \{ q_{i,j}^v [(1 - q_{i,j+1}^v - q_{i,j}^h - q_{i+1,j}^h)^2 \\ & + (1 - q_{i,j-1}^v - q_{i,j-1}^h - q_{i+1,j-1}^h)^2] \\ & + q_{i-1,j}^v [(1 - q_{i-1,j+1}^v - q_{i-1,j}^h - q_{i,j}^h)^2 \\ & + (1 - q_{i-1,j-1}^v - q_{i-1,j-1}^h - q_{i,j-1}^h)^2] \}, \end{aligned} \quad (12)$$

$$\begin{aligned} E_p^n = {} & C_p \sum_i \sum_j (q_{i,j}^n q_{i+1,j}^n + q_{i,j}^n q_{i-1,j}^n) \\ & + C_c \sum_i \sum_j q_{i,j}^n \\ & + C_l \sum_i \sum_j q_{i,j}^n [(1 - q_{i,j-1}^n - q_{i,j+1}^n)^2 \\ & + (1 - q_{i+1,j}^n - q_{i-1,j}^n)^2]. \end{aligned} \quad (13)$$

E_s^{ru}, E_s^{rt}, E_s^{lu}, E_s^{lt}: the functions to decide slanting edge parameters. They are defined as

$$E_s^{ru} = \sum_i \sum_j (1 - q_{i,j}^{ru}) |(x_{i,j} - x_{i,j+1}) \cdot (x_{i,j} - x_{i+1,j})|, \quad (14)$$

$$E_s^{rt} = \sum_i \sum_j (1 - q_{i,j}^{rt}) |(x_{i,j} - x_{i,j-1}) \cdot (x_{i,j} - x_{i-1,j})|, \quad (15)$$

$$E_s^{lu} = \sum_i \sum_j (1 - q_{i,j}^{lu}) |(x_{i,j} - x_{i,j-1}) \cdot (x_{i,j} - x_{i+1,j})|, \quad (16)$$

$$E_s^{lt} = \sum_i \sum_j (1 - q_{i,j}^{lt}) |(x_{i,j} - x_{i-1,j}) \cdot (x_{i,j} - x_{i,j+1})|. \quad (17)$$

E_p^{ru}, E_p^{rt}, E_p^{lu}, E_p^{lt}: the functions to restrict edge location and total number of slanting edges. They are defined as

$$\begin{aligned} E_p^{ru} = {} & C_p \sum_i \sum_j q_{i,j}^{ru} (q_{i,j+1}^{ru} + q_{i+1,j}^{ru} + q_{i-1,j}^{ru} + q_{i,j-1}^{ru} \\ & + q_{i+1,j+1}^{rt} + q_{i,j}^{rt} + q_{i,j+1}^{lu} + q_{i,j}^{lu} + q_{i,j}^{lt} + q_{i+1,j}^{lt}) \\ & + C_c \sum_i \sum_j q_{i,j}^{ru} \\ & + C_l \sum_i \sum_j q_{i,j}^{ru} [(1 - q_{i,j+1}^{rt})^2 + (1 - q_{i+1,j}^{rt})^2], \end{aligned} \quad (18)$$

$$\begin{aligned} E_p^{rt} = {} & C_p \sum_i \sum_j q_{i,j}^{rt} (q_{i-1,j}^{rt} + q_{i,j-1}^{rt} + q_{i,j+1}^{rt} + q_{i+1,j}^{rt} \\ & + q_{i-1,j-1}^{ru} + q_{i,j}^{ru} + q_{i-1,j}^{lu} + q_{i,j}^{lu} + q_{i,j}^{lt} + q_{i,j-1}^{lt}) \\ & + C_c \sum_i \sum_j q_{i,j}^{rt} \\ & + C_l \sum_i \sum_j q_{i,j}^{rt} [(1 - q_{i-1,j}^{ru})^2 + (1 - q_{i,j-1}^{ru})^2], \end{aligned} \quad (19)$$

$$\begin{aligned} E_p^{lu} = {} & C_p \sum_i \sum_j q_{i,j}^{lu} (q_{i,j-1}^{lu} + q_{i+1,j}^{lu} + q_{i-1,j}^{lu} + q_{i,j+1}^{lu} \\ & + q_{i+1,j-1}^{lt} + q_{i,j}^{lt} + q_{i,j-1}^{ru} + q_{i,j}^{ru} + q_{i,j}^{rt} + q_{i+1,j}^{rt}) \\ & + C_c \sum_i \sum_j q_{i,j}^{lu} \\ & + C_l \sum_i \sum_j q_{i,j}^{lu} [(1 - q_{i,j-1}^{lt})^2 + (1 - q_{i+1,j}^{lt})^2], \end{aligned} \quad (20)$$

$$\begin{aligned} E_p^{lt} = {} & C_p \sum_i \sum_j q_{i,j}^{lt} (q_{i-1,j}^{lt} + q_{i,j+1}^{lt} + q_{i+1,j}^{lt} + q_{i,j-1}^{lt} \\ & + q_{i-1,j+1}^{lu} + q_{i,j}^{lu} + q_{i-1,j}^{ru} + q_{i,j}^{ru} + q_{i,j}^{rt} + q_{i,j+1}^{rt}) \\ & + C_c \sum_i \sum_j q_{i,j}^{lt} \\ & + C_l \sum_i \sum_j q_{i,j}^{lt} [(1 - q_{i-1,j}^{lu})^2 + (1 - q_{i,j+1}^{lu})^2]. \end{aligned} \quad (21)$$

The coefficients C_p, C_c, C_l appearing in above equations are constants. They are determined during the neural network training.

In order to validate the effect of HNN and its application on image edge detection, an image shown in Fig. 4

(199×199 pixels, 8-bit gray level) is used for simulation. It is a training image of globular materials without noise.

Figure 5(a) shows the result of edge detection by Laplacian of Gaussian (LOG) operator. We can see that LOG operator is easy to lose edge details of image badly. Figure 5(b) shows the result of edge detection by Canny operator, this method is better than LOG, but it still loses some details. Figure 5(c) is obtained by Ref. [4] method, from the result we can find that this method lose some edge details at horizontal and vertical directions. Figure 5(d) shows the result of edge detection by the proposed method. From the results, it is verified that the proposed method can accurately detect the image of globular materials.

In order to detect anti-noise performance of HNN, the simulating experiments are performed for edge detection of a blur image. Figure 6(a) is a blur image generated by contaminating Fig. 4 with Gaussian white noise (equalizing value is zero, variance is 0.01). Figures 6(b) and (c) are results by operating a Canny operator and Ref. [4] method, respectively. Figure 6(d) represents the result by the proposed method. Comparing these methods,

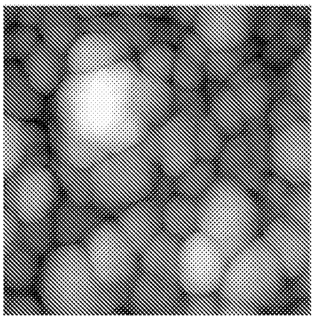

Fig. 4. A testing image without noise.

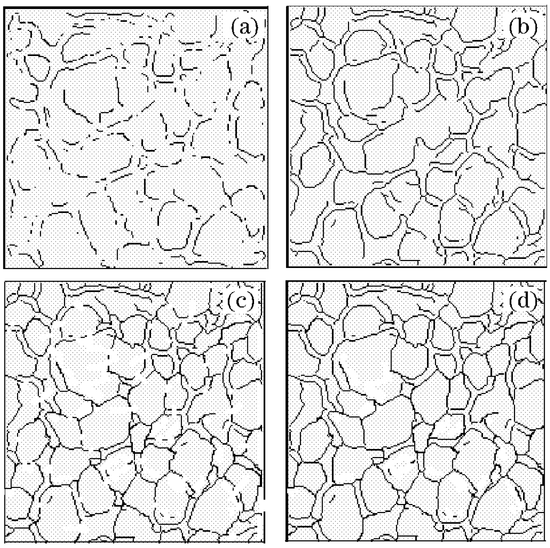

Fig. 5. The results of edge detection. (a) Using LOG operator; (b) using Canny operator; (c) using Ref. [4] method; (d) using the proposed method.

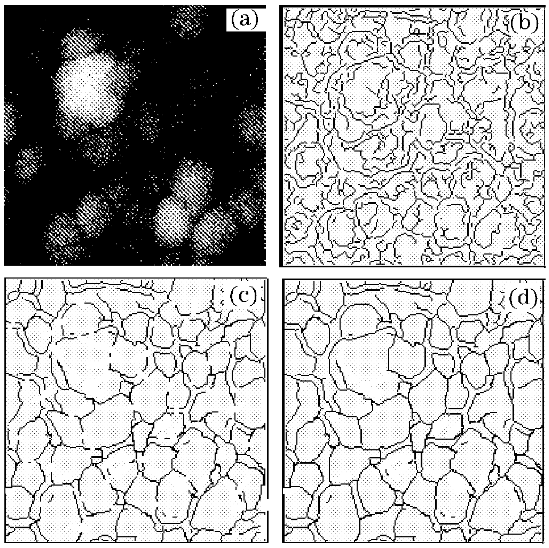

Fig. 6. (a) Blur image; (b) result by Canny operator method; (c) result by Ref. [4] method; (d) result by the proposed method.

we can see that the proposed method based on HNN can detect the edges more accurately than the conventional methods.

In conclusion, this paper proposes the edge detection method using a HNN. According to the characteristics of image edge detection, new edge parameters are set and express the image edge details. Energy function is founded using these parameters. Tests on the image edge detection show that the proposed method has good edge closeness and accurate edge, at the same time it has better anti-noise performance. The proposed method can detect the edges of blur images more accurately than the conventional methods.

This work was supported by the Foundation of Shanghai Technology Committee under Grant No. 021111125. Y. Zhang's e-mail address is zyh_sj2002@sjtu.edu.cn.

References

1. D. C. Marr and E. Hildreth, C. Proc. Roy. Soc. **B207,** 283 (1980).
2. J. Canny, IEEE Trans. PAMI **8,** 679 (1986).
3. M. Muneyasu, K. Hotta, and T. Hinamoto, in *Transaction of Neural Networks on IEEE International Conference* **4,** 1703 (1995).
4. Y. Uchiyama, M. Haseyama, and H. Kitajima, in *Transaction of 2001 IEEE International Symposium on Circuits and Systems* **3,** 608 (2001).
5. M. Muneyasu, K. Yamamote, and T. Hinamoto, in *Transaction of International Conference on Image Processing* **2,** 33 (1995).
6. Y. T. Wang, W. Q. Yuan, J. C. Zhou, and S. Wang, J. of Northeastern University (in Chinese) **20,** 126 (1999).
7. H. J. Yang and D. Q. Liang, Acta Electron. Sin. (in Chinese) **29,** 51 (2001).