Soft Comput (2007) 11: 221–228
DOI 10.1007/s00500-006-0064-8

FOCUS

Zeng-Guang Hou · Ki-Young Song
Madan M. Gupta · Min Tan

Neural units with higher-order synaptic operations for robotic image processing applications

Published online: 18 March 2006

Abstract Neural units with higher-order synaptic operations have good computational properties in information processing and control applications. This paper presents neural units with higher-order synaptic operations for visual image processing applications. We use the neural units with higher-order synaptic operations for edge detection and employ the Hough transform to process the edge detection results. The edge detection method based on the neural unit with higher-order synaptic operations has been applied to solve routing problems of mobile robots. Simulation results show that the proposed neural units with higher-order synaptic operations are efficient for image processing and routing applications of mobile robots.

1 Introduction

A mobile robot requires multiple sensors such as CCD camera, ultrasonic and infrared sensors to locate its position, to avoid obstacles, and to find an optimal path leading to the destination. The robot fuses all the information acquired from these sensors and makes decisions to control its direction and speed. Among these sensors, the CCD camera serves as a visual sensor for the decision, especially for the wide range decision, of mobile robot routing operations. The most important task for the CCD camera and the processor is to capture and process images, and to discriminate the obstacles in the paths. Edge detection is a simple while efficient method for mobile robot routing problems. An edge is where the vertical and the horizontal surfaces of objects meet. For a mobile robot, an edge may mean where the path ends and the wall or obstacle starts, or where the paths and obstacles intersect. Traditionally, edges have been defined loosely as pixel intensity discontinuities within an image. Usually it is easy to detect edges with high contrast, or those with high S/N ratio. But in changing light or dim circumstances, the edge detection may not be so easy. In the mean time, the on-board CPU should recognize the feasible paths and path junctions or crossings using the edge features.

In this paper, we address edge detection problems of vision-based routing for mobile robots using higher-order neural units, i.e., neural units with higher-order synaptic operations, as an image processing tool. Simulation results show that these proposed neural units with higher-order synaptic operations are effective and efficient for image processing and mobile robot routing operations.

2 Neural units with higher-order synaptic operation

Most neural networks employ linear neurons or neural units, which have a linear correlation between the input vector and the synaptic weight vector. In recent years, neural units with higher-order synaptic operations and their composed neural networks, where the higher-order nonlinear correlation manipulations are used, have been proved to have good computational, storage, pattern recognition, and learning properties, and can be implemented in hardware [24]. These networks satisfy the Stone–Weierstrass theorem and hence considered to improve the approximation and generalization capabilities of the network. Extensive research attempts have been made by Rumelhart et al. [18,19], Giles and Maxwell [4], Softky and Kammen [23], Reid et al. [17], Samad and Harper [20], and Homma and Gupta [8], Ghosh and Shin [3], Gupta et al. [6], Redlapalli [15,16] to develop neural units with higher-order synaptic operations for different applications. In the literature of adaptive neural control, neural networks are primarily used as on-line approximators for the unknown nonlinearities due to their inherent approximation capabilities. However, the conventional models of neurons

Z.-G. Hou (✉) · M. Tan
Laboratory of Complex Systems and Intelligence Science,
Institute of Automation, Chinese Academy of Sciences,
Beijing 100080, China
E-mail: zengguang.hou@ia.ac.cn

K.-Y. Song · M.M. Gupta
Intelligent Systems Research Laboratory,
College of Engineering, University of Saskatchewan,
Saskatoon, Saskatchewan, S7N 5A9, Canada
E-mail: madan.gupta@usask.ca

cannot deal with the discontinuities in the input training data. In an effort to overcome such limitations of conventional neural units with linear synaptic operations, some researchers have turned their attention to neural units with higher-order synaptic operations [6].

It is to be noted that neural units with higher-order synaptic operations contain all the linear and nonlinear correlations terms of the input components. A generalized structure of neural units with higher-order synaptic operations is a polynomial network that includes the weighted sums of products of selected input components with an appropriate power. This type of network is called a sigma–pi network [18,19]. The synaptic operation of the sigma–pi network creates the product of the selected input components computed with power operation while the conventional neural units compute the synaptic operation as a weighted sum of all the neural inputs. Since sigma–pi networks result in exponential increase in the number of parameters, some modified forms of sigma–pi networks were introduced by Shin and Ghosh [22] which involve smaller number of weights than the neural units with higher-order synaptic operations. They are called as pi–sigma network where the synaptic operation is the product of the weighted sum of all nonlinear correlations terms of the input components. Another type of higher-order neural network called the ridge polynomial neural network was introduced by Shin and Ghosh [22]. However, the problem encountered with these neural networks is the combinatorial increase of the weight numbers; that is, as the input size increases, the number of weights in higher-order neural networks increases exponentially [7]. Some papers proposed higher-order neural networks based on the Hopfield type or recurrent neural networks [9,11–13,20,21], which has been used for solving various optimization problems and thus is not the case to be discussed in this paper.

2.1 Neural units with quadratic synaptic operation

The synaptic operation of the neural unit with quadratic synaptic operation embraces both the first- and second-order neural input combinations with the synaptic weights. A neural unit with quadratic synaptic operation and with n scalar inputs is defined as follows:

$$\begin{aligned} z &= \sum_{i=0}^{n}\sum_{j=i}^{n} w_{ij}x_ix_j \\ &= w_{00}x_0x_0 + w_{01}x_0x_1 + \cdots + w_{n(n-1)}x_nx_{n-1} + w_{nn}x_nx_n, \end{aligned} \tag{1}$$

where w_{00} is the threshold (bias) weight and $x_0 = 1$ is the constant bias. If we consider the n scalar neural inputs and the constant bias input $x_0 = 1$ as an augmented $(n+1)$-dimensional input vector defined by $\mathbf{x}_a = \{1, x_1, \ldots, x_n\}^{\mathrm{T}} \in R^{n+1}$, Eq. (1) can be expressed as

$$z = \mathbf{x}_{\mathrm{a}}^{\mathrm{T}}\mathbf{W}\mathbf{x}_{\mathrm{a}}, \tag{2}$$

where $\mathbf{W}$ is the augmented synaptic weight matrix for the neural unit with quadratic synaptic operation, and is defined by

$$\mathbf{W} = \begin{bmatrix} w_{00} & w_{01} & \cdots & w_{0n} \\ 0 & w_{11} & \cdots & w_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{nn} \end{bmatrix} \in R^{(n+1)(n+1)}. \tag{3}$$

The weights w_{ij} and w_{ji}, $i, j \in \{0, 1, 2, \ldots, n\}$, in the augmented matrix $\mathbf{W}$ yield the same quadratic term x_ix_j or x_jx_i. Therefore, an upper triangle or a lower triangle of the augmented weight matrix $\mathbf{W}$ is sufficient to describe the discriminant function that assigns a measurement value to very point in the input space. The upper triangle matrix gives the general quadratic discriminant function as

$$z = \mathbf{x}_{\mathrm{a}}^{\mathrm{T}}\mathbf{W}\mathbf{x}_{\mathrm{a}} = \sum_{i=0}^{n}\sum_{j=i}^{n} w_{ij}x_ix_j \in R, \quad x_0 = 1. \tag{4}$$

The total number of weights involved in this structure is $(n+1)(n+2)/2$, where n is the number of inputs.

A further observation of the augmented weight matrix W reveals that the conventional neural units with linear synaptic operation are a subset of the neural unit with quadratic synaptic operation. For example, the first row of the weight matrix $\mathbf{W} = [w_{00}\ w_{01}\ w_{02}\ \cdots\ w_{0n}] \in R^{n+1}$ can produce the weighted linear combination of the neural inputs:

$$z = \sum_{j=0}^{n} w_{0j}x_j. \tag{5}$$

The output of the neural unit with quadratic synaptic operation is given by

$$y = \phi(z) \in R, \tag{6}$$

where $\phi(\cdot)$ is the activation function of the neural unit, which defines the somatic operation.

2.2 Neural units with cubic synaptic operation

A neural unit with cubic synaptic operation is given by

$$\begin{aligned} z &= \sum_{i=0}^{n}\sum_{j=i}^{n}\sum_{k=j}^{n} w_{ijk}x_ix_jx_k \\ &= w_{000}x_0x_0x_0 + w_{001}x_0x_0x_1 + w_{002}x_0x_0x_2 \\ &\quad + \cdots + w_{nn(n-1)}x_nx_nx_{n-1} + w_{nnn}x_nx_nx_n, \end{aligned} \tag{7}$$

where $x_0 = 1$ is the threshold, and $\mathbf{x}_{\mathrm{a}} = [x_0, x_1, x_2, \ldots, x_{n-1}, x_n]^{\mathrm{T}}$ is the vector of augmented neural inputs. The output of the somatic operation is defined by a nonlinear mapping $y = \phi(z)$.

Recall that $x_0 = 1$, and simplify the subscripts of the weights in Eq. (7) if their first digits begin with 0s, for example, w_{000} and w_{001} are, respectively, represented by w_0 and w_1. Then the neural unit with cubic synaptic operation and with two neural inputs ($n = 2$) is defined by

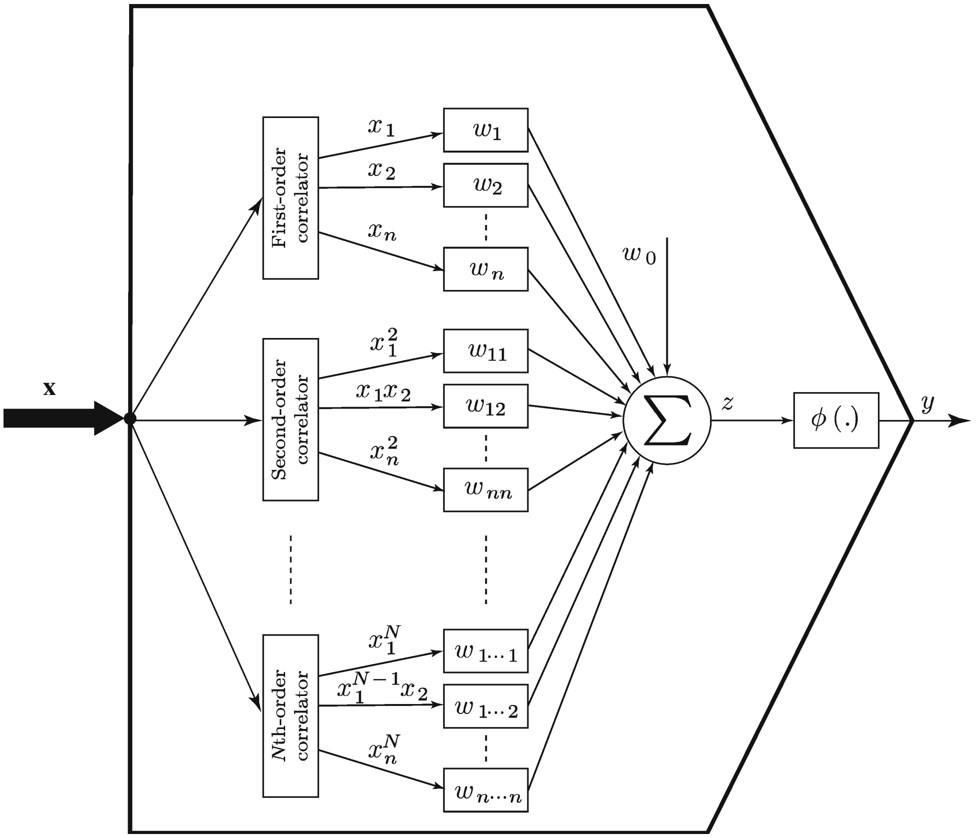

Fig. 1 The neural unit with higher-order synaptic operation

$$
\begin{aligned}
y = \phi \big(&w_0 + w_1x_1 + w_2x_2 + w_{11}x_1^2 + w_{12}x_1x_2 + w_{22}x_2^2 \\
&+w_{111}x_1^3 + w_{112}x_1^2x_2 + w_{122}x_1x_2^2 + w_{222}x_2^3\big). \quad (8)
\end{aligned}
$$

2.3 Neural units with the Nth order synaptic operation

A neural unit with the Nth-order synaptic operation and with an n-dimensional input vector can be expressed as

$$y = \phi(z), \quad (9)$$

$$
\begin{aligned}
z = w_0 + &\sum_{i_1}^{n} w_{i_1}x_{i_1} + \sum_{i_1,i_2}^{n} w_{i_1i_2}x_{i_1}x_{i_2} + \cdots \\
&+ \sum_{i_1,\ldots,i_N}^{n} w_{i_1\ldots i_N}x_{i_1}\cdots x_{i_N} \quad (10)
\end{aligned}
$$

The structure of the neural unit with higher-order synaptic operation is shown in Fig. 1, where $\mathbf{x} = [x_1, x_2, \ldots, x_n] \in R^n$ is the vector of neural inputs, $y \in R$ is an output scalar, and $\phi(\cdot)$ is a strictly monotonic activation function such as a sigmoidal function whose inverse, $\phi^{-1}(\cdot)$, exists.

The neural units with higher-order synaptic operation can be used in conventional feedforward neural network structures as the hidden units to form neural networks. In this case, however, consideration of the higher correlation may improve the approximation and generalization of the neural networks.

It has been reported in our previous work (See Refs. [8, 15] for more details) that these neural units have high efficiency for signal processing and control problems.

3 Edge detection using neural units with higher-order synaptic operation and applications to mobile robot routing problems

Traditional edge detection and enhancement techniques include Sobel, Laplace and Gabor operators, etc [2]. More recently, a few tentative researches were published using neural networks for efficient processing of digital images [1,14, 27]. Most of them are based on the multi-layer feedforward network architectures, which usually have longer learning time. In our present research, we developed an approach on the basis of a neural unit with higher-order synaptic operations by which using one unit is capable of dealing with complicated problems.

The Gaussian function is now applied as the discriminant function because the Gaussian function performs the mean of the time instance values during a period with respect to the activated points. Moreover, the Gaussian function is continuous and differentiable, so that this function should be taken as a discriminant function.

This procedure for the neural edge detection is shown as follows and illustrated in Fig. 2:

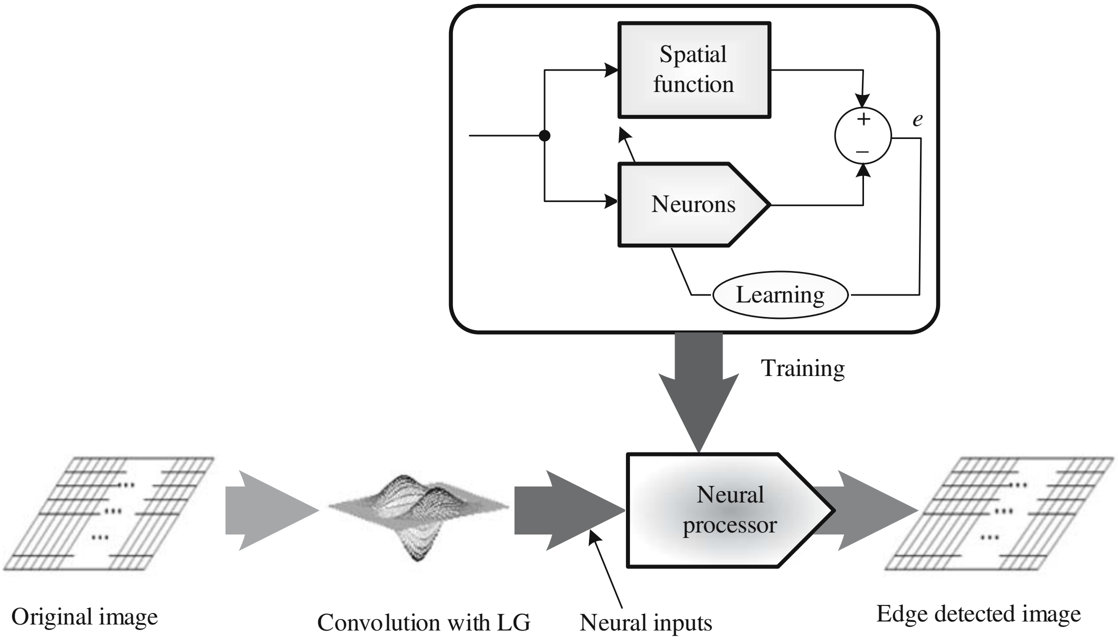

Fig. 2 Neural procedure for edge detection: the convoluted image is processed through the trained neural processor with the spatial function

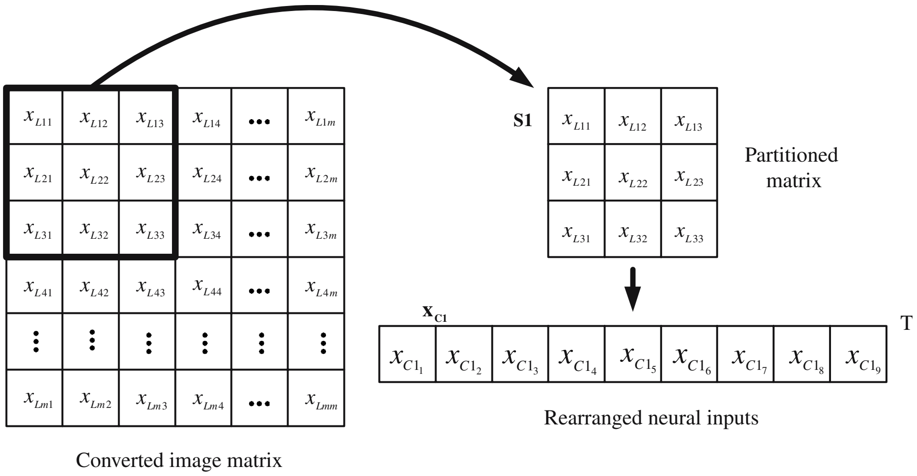
x_{L11} x_{L12} x_{L13} x_{L14} ••• x_{L1m}
x_{L21} x_{L22} x_{L23} x_{L24} ••• x_{L2m}
x_{L31} x_{L32} x_{L33} x_{L34} ••• x_{L3m}
x_{L41} x_{L42} x_{L43} x_{L44} ••• x_{L4m}
x_{Lm1} x_{Lm2} x_{Lm3} x_{Lm4} ••• x_{Lmm}
Converted image matrix
S1
x_{L11} x_{L12} x_{L13}
x_{L21} x_{L22} x_{L23}
x_{L31} x_{L32} x_{L33}
Partitioned matrix
$\mathbf{x}_{C1}$
T
x_{C1_1} x_{C1_2} x_{C1_3} x_{C1_4} x_{C1_5} x_{C1_6} x_{C1_7} x_{C1_8} x_{C1_9}
Rearranged neural inputs

Fig. 3 Neural input matrix from the convoluted image matrix

- To take an image performing as the eyes
- To apply the second-order differentiation of the Gaussian function called the *Laplacian of the Gaussian* (LG) to process the captured image
- To teach the neural units with color spatial functions
- To detect the edges of the objects in the image

The Gaussian function is defined as

$$Ga(x) = \mathrm{e}^{-\alpha x^2}, \tag{11}$$

where α is the slope rate of the Gaussian function. The first-order differentiation of the Gaussian function is derived as

$$\mathrm{DG}(x) = -2\alpha x \mathrm{e}^{-\alpha x^2} = -2\alpha x Ga(x). \tag{12}$$

Thus, the LG is derived as

$$\begin{aligned}\mathrm{LG}(x) &= \{-2\alpha x Ga(x)\}' \\ &= (-2\alpha x)' Ga(x) + (-2\alpha x) Ga'(x) \\ &= (-2\alpha) Ga(x) + (-2\alpha x)\{-2\alpha x Ga(x)\} \\ &= (-2\alpha + 4\alpha^2 x^2) Ga(x) \\ &= 2\alpha(2\alpha x^2 - 1) Ga(x).\end{aligned} \tag{13}$$

The three-dimensional LG is applied as an optimal aggregation function. The captured image is convoluted by the LG whose size for the suitable experiment is (5×5). The convoluted data of the image is fed to the neural inputs. The neural processor is constructed by training the static neurons with the color spatial function. The spatial function verifies the contrast or brightness of the color in the image. The range of

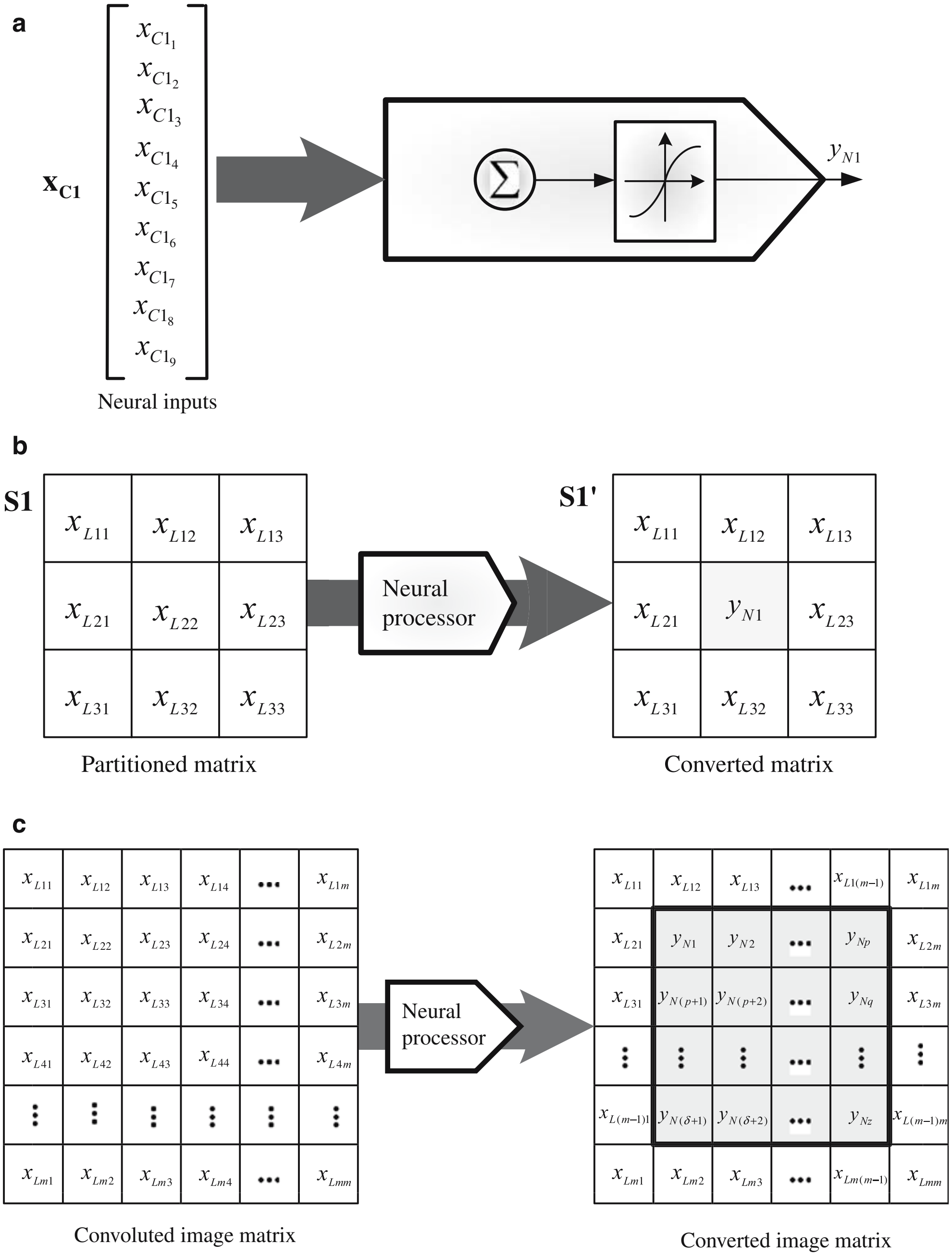
a
$\mathbf{x}_{C1}$
x_{C1_1}
x_{C1_2}
x_{C1_3}
x_{C1_4}
x_{C1_5}
x_{C1_6}
x_{C1_7}
x_{C1_8}
x_{C1_9}
Neural inputs
Σ
y_{N1}
b
S1
x_{L11} x_{L12} x_{L13}
x_{L21} x_{L22} x_{L23}
x_{L31} x_{L32} x_{L33}
Partitioned matrix
Neural processor
S1'
x_{L11} x_{L12} x_{L13}
x_{L21} y_{N1} x_{L23}
x_{L31} x_{L32} x_{L33}
Converted matrix
c
x_{L11} x_{L12} x_{L13} x_{L14} ••• x_{L1m}
x_{L21} x_{L22} x_{L23} x_{L24} ••• x_{L2m}
x_{L31} x_{L32} x_{L33} x_{L34} ••• x_{L3m}
x_{L41} x_{L42} x_{L43} x_{L44} ••• x_{L4m}
x_{Lm1} x_{Lm2} x_{Lm3} x_{Lm4} ••• x_{Lmm}
Convoluted image matrix
Neural processor
x_{L11} x_{L12} x_{L13} ••• $x_{L1(m-1)}$ x_{L1m}
x_{L21} y_{N1} y_{N2} ••• y_{Np} x_{L2m}
x_{L31} $y_{N(p+1)}$ $y_{N(p+2)}$ ••• y_{Nq} x_{L3m}
$x_{L(m-1)1}$ $y_{N(\delta+1)}$ $y_{N(\delta+2)}$ ••• y_{Nz} $x_{L(m-1)m}$
x_{Lm1} x_{Lm2} x_{Lm3} ••• $x_{Lm(m-1)}$ x_{Lmm}
Converted image matrix

Fig. 4 The neural processor and conversion of the image matrix. **a** Neural processor; **b** Conversion of the selected image matrix; **c** Conversion of the entire image matrix

the spatial function is from 0 to 255, which signifies the gray level color.

The neural processor generates a one-dimensional neural output with a 3×3 neural input matrix from the convoluted image. Each neural output affects the neighboring segments charging or discharging the illumination. The effect produces the clear edge detected image. Figure 3 shows the transformation of the image serving as inputs to the neural processor.

The partitioned matrix is then rearranged as the neural inputs in the order of the elements in rows and columns. The selected 3×3 matrix S1 becomes the 9×1 neural input x_{C1}. Each element of x_{C1} is processed by the neural processor. After the synaptic and somatic operations, the neural processor generates the neural output as

$$y_{N1}(x) = \phi\left[v_{a1}(\mathbf{x})\right], \tag{14}$$

where $v_{a1}(\cdot)$ denotes the higher-order synaptic operation. After the first selected matrix procedure, x_{L22} of S1 is replaced with y_{N1} because the middle element of S1 is adjacent to every other element. Thus, S1 is renamed as S1′. Figure 4 illustrates the neural processing of the image matrix.

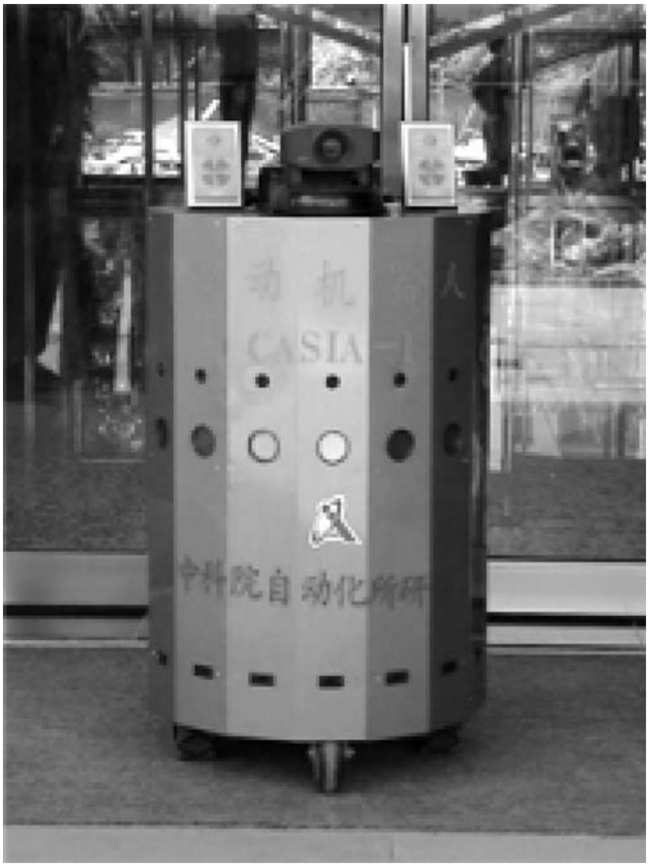

Fig. 5 The mobile robot

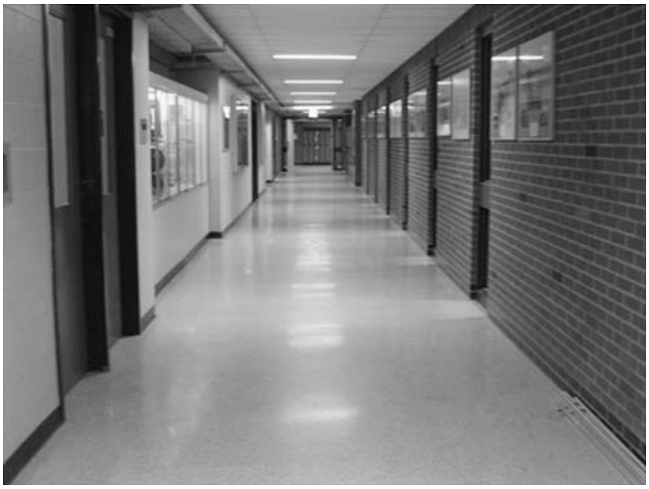

Fig. 6 Original image of a corridor

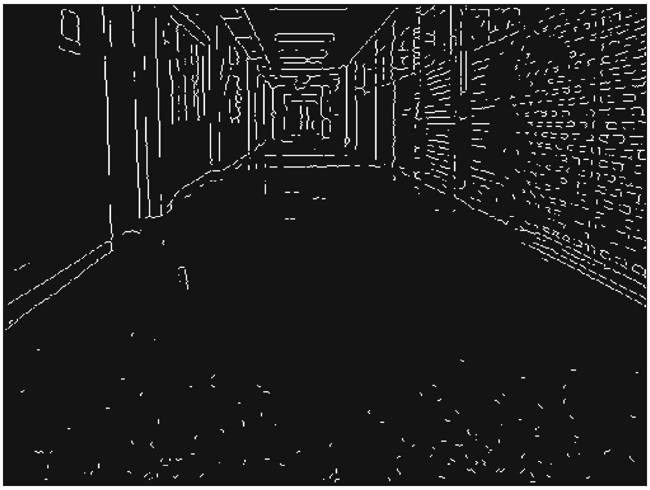

Fig. 7 Edge detection using higher-order neural units

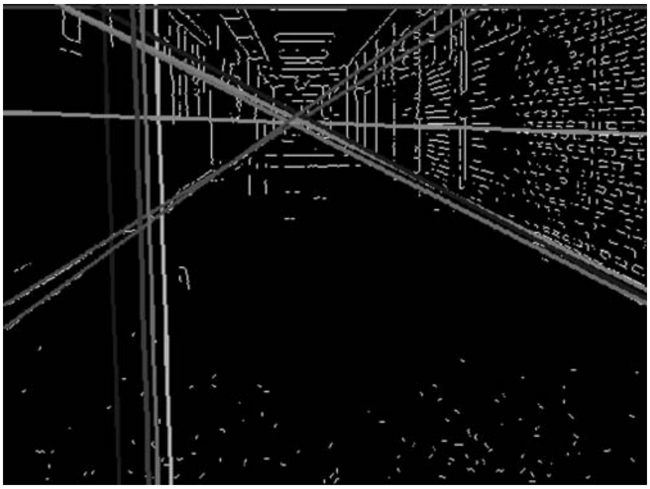

Fig. 8 Result of the Hough transform with the threshold being 0.7

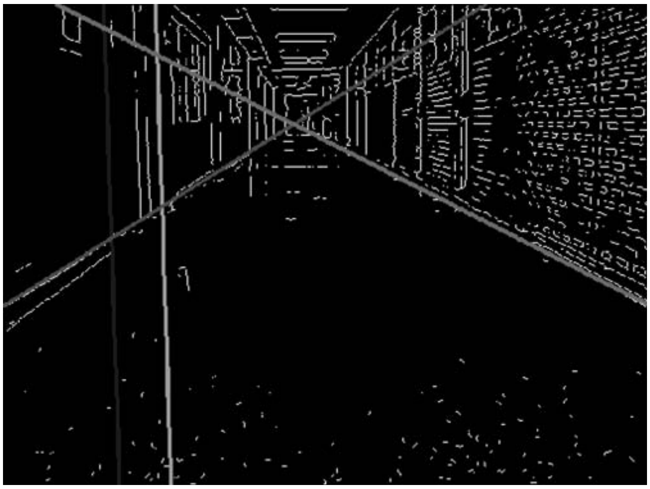

Fig. 9 Result of the Hough transform with the threshold being 0.8

Since the new converted matrix is replaced the first partitioned matrix, the next partitioned matrix contains the new computed element from the previous neural processed results, and so forth. After the entire neural procedure is completed, the convoluted image matrix is obtained.

The transformed convoluted image after the neural processor enhances the edges of the objects in the given image by the higher-order computation. In order to obtain the optimal edge detected image, the slope rate of the Gaussian function and the size of the sectioned matrix should be adjustable.

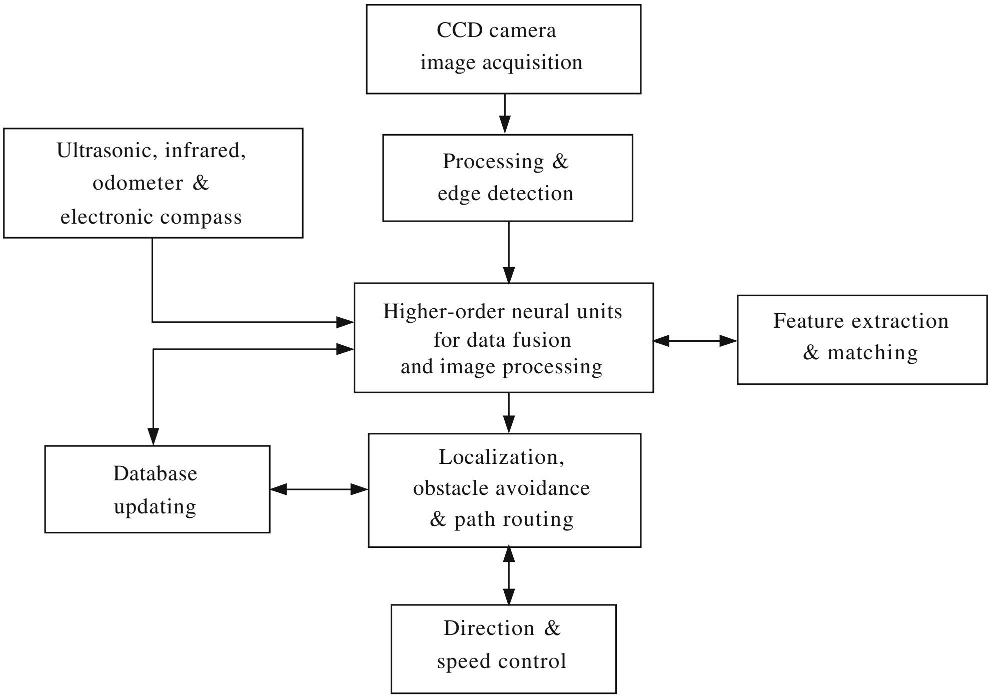

Fig. 10 Block diagram of information processing for the robot system

In the field of computer vision and image processing, it is imperative to detect the basic shapes such as lines and circles. A higher-order neural unit based edge detection method was developed in this paper to find out the shapes of the images. However, the neural edge detection alone cannot facilitate the machine to recognize the objects. In order to employ the edge detection results, Hough transform is required to compute and transfer the image data to the gadget. Hough transform is one of the powerful methods which detect the basic shapes from the landmarks even though some landmarks are failed or covered up. The original form of Hough transform involved parameterizing lines by the following slope–intercept equation:

$$y = mx + c. \tag{15}$$

Every point on a straight edge of the edge detected image is plotted as a line in (m, c) space corresponding to all the (m, c) values consistent with its coordinates, and lines are detected in this space. The detrimental of the original Hough transform is that the value of m or c becomes infinity if the line is parallel with x-axis or y-axis. The modified Hough transform was used in order to remove this disadvantage, which substitutes the normal (θ, ρ) form for the slope–intercept format for the straight line as

$$\rho = x \cos \theta + y \sin \theta. \tag{16}$$

The set of lines passing through a point is represented as a set of sine curves in (θ, ρ) space. Then multiple hits in (θ, ρ) space signify the presence of lines in the original image. Consider that some lines are detected from the landmarks. The parameter space, i.e., (θ, ρ) space, is divided into many small subspaces, and the relationships vote for the corresponding subspace. The central point values of the subspaces which have many votes are selected as the parameters of the lines to be detected. The lines are detected corresponding to the threshold. The image captured with the CCD camera is processed by the neural edge detector and Hough transform.

Using the computed results, the mobile robot, which is shown in Fig. 5, can obtain the data of the edge lines of the corridor, and travel along the corridor and avoid colliding with obstacles. The detected edges of corridors can be used for mobile robot routing and navigation decisions. Ultrasonic and infrared sensors usually serve for real-time obstacle avoidance. The visual system of the mobile robot serves for long-term routing applications of the robot. An original image captured with the CCD camera of the robot is shown in Fig. 6. The neuro-vision system detects the edges of the image and identifies the lines of the corridors and the junctions or crossing. Figure 7 shows the edge detection result of the corridor image, and Figs. 8 and 9 illustrate the straight lines with different thresholds, along which the mobile robot should move.

The primary simulation results showed that this method is of great potential for practical use, and worth further studies. We have developed a vision-based decision-making unit, which is shown in Fig. 10, for our developed mobile robot using the proposed neural units with higher-order synaptic operations. We are also developing dynamic neural units for wider applications in the field of information processing and control for robotics [10,25,26].

4 Conclusions

Using the neural units with higher-order synaptic operation, we present a scheme for visual image processing of mobile robots. We first employ the neural units with higher-order synaptic operation to obtain the edge features of an image, and then use the Hough transform to process the edge detection result. Simulation studies showed that the proposed method is efficient for mobile robot vision system and routing applications.

Acknowledgements This research is supported in part by the National Natural Science Foundation of China (Grant Nos. 60205004, 60334020 and 50475179), the National Basic Research Program (973) of China (Grant No. 2002CB312200), the Hi-Tech R&D Program (863) of China (Grant Nos. 2002AA423160 and 2005AA420040), the K.C. Wong Education Foundation of Hong Kong, the Science and Technology New Star Program of Beijing (Grant No. H020820780130), and the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

1. Barsi A, Heipke C, Willrich F (2002) Junction extraction by artificial neural network system - JEANS. In: IntArchPhRS Com.III, Graz, vol XXXIV, Part 3b, pp 18–21
2. Davies ER (1990) Machine vision: Theory, Algorithms, Practicalities. Academic Press, San Diego
3. Ghosh J, Shin Y (1992) Efficient higher order neural networks for classification and function approximation. Int J Neural Syst 3(4):323–350
4. Giles CL, Maxwell T (1987) Learning, invariance, and generalization in high-order neural networks. Appl Optics 26(23):4972–4978
5. Gupta MM, Knopf GK (1994) Neuro-vision systems: principles and applications. IEEE Press, New York
6. Gupta MM, Jin L, Homma N (2003) Static and dynamic neural networks: from fundamentals to advanced theory. Wiley/IEEE Press, New York
7. He Z, Siyal MY (1998) Modification on higher-order neural networks. In: Proceedings of the artificial networks in engineering Conference, vol 8, pp 31–36
8. Homma N, Gupta MM (2002) Superimposing learning for backpropagation neural networks. Bull Coll Med Sci, Tohoku Univ, Jpn 11(2):253–259
9. Hou ZG (2001) A hierarchical optimization neural network for large-scale dynamic systems. Automatica 37(12):1931–1940
10. Hou ZG (2005) Principal component analysis (PCA) for data fusion and navigation of mobile robots. In: Kantor P et al (eds) Springer lecture notes in computer science (LNCS): intelligence and security informatics, vol 3495. Springer, Berlin Heidelberg New York, pp 610–611
11. Hou ZG, Wu C, Bao P (1998) A neural network for hierarchical optimization of nonlinear large-scale systems. Int J Syst Sci 29(2):159–166
12. Jin L, Gupta MM (1999) Stable dynamic backpropagation learning in recurrent Neural Networks. IEEE Trans Neural Netw 10(6):1321–1334
13. Joya G, Atencia MA, Sandoval F (1997) Associating arbitrary-order energy functions to an artificial neural network, implications concerning the resolution of optimization problems. Neurocomputing 14:139–156
14. Moradi S (2002) Victim detection with infrared camera in a ‘rescue robot’. In: Proceedings of the international congress on autonomous intelligent systems (ICAIS), Australia
15. Redlapalli SK (2004) Development of neural units with higher-order synaptic operations and their applications to logic circuits and control problems. Master’s thesis, Department of Mechanical Engineering, University of Saskatchewan
16. Redlapalli SK, Gupta MM, Song KY (2003) Development of quadratic neural unit with applications to pattern classification. In: Proceedings of the fourth international symposium on uncertainty, modeling and analysis. College Park, Maryland, pp 141–146
17. Reid MB, Spirskovka L, Ochoa E (1989) Rapid training of higher-order neural networks for invariant pattern recognition. In: Proceedings of international joint conference on neural networks, vol 3, pp 689–692
18. Rumelhart DE, McClelland JL (1986) Parallel distributed processing: explorations in the microstructure of cognition. MIT Press, Cambridge
19. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: Rumelhart DE, McCleand JL (eds) Parallel distributed processing, vol 1. MIT Press, Cambridge, pp 318-362
20. Samad T, Harper P (1990) High-order Hopfield and Tank optimization networks. Parallel Comput 16:287–292
21. Sejnowski TJ (1986) Higher-order Boltzmann machines. In: Denker JS (ed) Neural networks for computing. American Institute of Physics, New York, pp 398–403
22. Shin Y, Ghosh J (1991) The pi–sigma network: an efficient higher-order network for pattem classification and function approximation. In: Proceedings of international joint conference on neural networks, Seattle, vol I, pp 13–18
23. Softky RW, Kammmen DM (1991) Correlations in high dimensional or asymmetrical data sets: Hebbian neuronal processing. Neural Netw 4(3):337–347
24. Taylor JG, Commbes S (1993) Learning higher-order correlations. Neural Netw 6(3):423–428
25. Zou A, Hou ZG, Tan M (2005) Support vector machines for color image segmentation with applications to mobile robot localization problems. In: Huang DS et al (eds) Springer lecture notes in computer science (LNCS): advances in intelligent computing, vol 3645. Springer, Berlin Heidelberg New York, pp 443–452
26. Zou A, Hou ZG, Zhang L, Tan M (2005) A neural network-based camera calibration method for mobile robot localization problems. In: Wang J, Liao X, Yi Z (eds) Springer lecture notes in computer sciences (LNCS): advances in neural networks, vol 3498. Springer, Berlin Heidelberg New York, pp 277–284
27. van der Zwaag BJ, Slump K (2002) Analysis of neural networks for edge detection. In: Proceedings of the ProRISC workshop on circuits, systems and signal processing, Netherlands, pp 580–586