# Relaxation Labeling Using an Improved Hopfield Neural Network

Long Cheng, Zeng-Guang Hou, and Min Tan

Key Laboratory of Complex Systems and Intelligence Science,
Institute of Automation, The Chinese Academy of Sciences,
P.O. Box 2728, Beijing 100080, China
{long.cheng, zengguang.hou, min.tan}@ia.ac.cn

**Abstract.** The relaxation labeling is a useful technique to deal with local ambiguity and achieve consistency. In [1], some useful comments indicate several common properties exist in the relaxation process and the neural network technique. Neural networks can be used as an efficient tool to optimize the average local consistency function whose optimal solution results in a compatible label assignment. However, most of current investigations in this field are based on the standard Hopfield neural network (SHNN) presented in [2]. In this paper, an improved Hopfield neural network (IHNN) presented in [3] is utilized to fulfill relaxation labeling. Compared to the SHNN, this approach has some advantages. 1) The IHNN uses fewer neurons than that of SHNN. 2) The activation function of IHNN is easier to be implemented than that of SHNN. 3) The IHNN does not contain any penalty parameters. It can generate the exact optimal solution. Some experimental results illustrate that the IHNN approach can obtain a better labeling performance than that of SHNN.

## 1 Introduction

The relaxation labeling is a useful technique to deal with local ambiguity and achieve consistency. It has far broader applications such as scene labeling, curve detection and enhancement, and image segmentation. There are basically three types of relaxation processes, namely discrete relaxation, fuzzy relaxation, and probabilistic relaxation [4]. In this paper, neural network technique is utilized to fulfill the probabilistic relaxation whose general idea is described as follows. This relaxation process involves a set of $n$ objects $O_1, O_2, \cdots, O_n$ and a set of $m$ labels $C_1, C_2, \cdots, C_m$. For each object $O_i$, a measurement $p_{ij}$ is used as the probability of the case that $O_i$ has the label $C_j$. For each $i \in \{1, 2, \cdots, n\}$, $p_{ij}$ has to satisfy the following condition

$$\begin{gathered}\sum_{j=1}^{m} p_{ij} = 1, \\ 0 \le p_{ij} \le 1, \quad \forall j = 1, 2, \cdots, m.\end{gathered} \tag{1}$$

Suppose further that the label assignment is correlated. For example, label $C_j$ at object $O_i$ can influence label $C_k$ at object $O_h$. This compatibility of the pair of label assignments $O_i \in C_j$ and $O_h \in C_k$ can be quantitatively measured by $r(i,j;h,k)$. If object $O_i$ having label $C_j$ tends to support the object $O_h$ with label $C_k$, then $r(i,j;h,k)$ is positive and larger, vice versa.

To apply relaxation labeling, each object is firstly assigned the initial probability $p_{ij}^{(0)}$ of its possible label memberships. Then relaxation process uses the compatibility measurement $r(i,j;h,k)$ to find a set of $n$ classifications of all the objects which are as compatible as possible. Rosenfeld et al. proposed an iterative updating algorithm to find this compatible classification [5]. Specifically, for each object and each label in the $r$th iteration, one computes the $q_{ij}^{(r)}$ by following equation

$$q_{ij}^{(r)} = \frac{1}{n}\sum_{h=1}^{n}\sum_{k=1}^{m} r(i,j;h,k)p_{hk}^{(r)}. \tag{2}$$

$q_{ij}^{(r)}$ denotes the contribution from the other objects to support the assignment $p_{ij}^{(r)}$. Then the new assignment value is adjusted according to following equation

$$p_{ij}^{(r+1)} = \frac{p_{ij}^{(r)}[1+q_{ij}^{(r)}]}{\sum\limits_{k=1}^{m} p_{ik}^{(r)}[1+q_{ik}^{(r)}]}. \tag{3}$$

In [7], Hummel and Zucker proposed following quantity criterion named average local consistency function to evaluate the performance of relaxation labeling.

$$Z(p) = \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}\left(\sum_{h=1}^{n}\sum_{k=1}^{m} r(i,j;h,k)p_{hk}\right) \tag{4}$$

They demonstrated that above hoc updating formula (2) and (3) are just an approximate approach to find the maximal solution to average local consistency function. Thus the probabilistic relaxation problem can be converted to a constrained optimization problem. In this point of view, many traditional numerical optimization approaches such as gradient ascent method can be used to fulfill relaxation process [7]. However these traditional methods suffer from their enormous computation cost and cannot satisfy the real-time processing requirement when the relaxation problem is complicated. So new approaches are required to reduce algorithm complexity and increase computational efficiency.

In recent years, neural networks as an efficient optimization tool have attracted a lot of attention in scientific and engineering applications. Rather than solving problems by numerical iteration, the fundamental idea behind the neural approach to optimization problems is: by setting up a neural circuit which is determined by a set of differential equations, for some initial network state, the neural networks will eventually evolve to an equilibrium state and this state will coincide with the optimal solution to the original problem. Since Hopfield and Tank's classic work [2], various neural network approaches have been proposed to resolve many kinds of optimization problems [3], [8], [9] and [10]. Some researchers also utilized neural network approach to fulfill relaxation process [1],

[11] and [12]. In [1], some useful comments indicate several common properties exist in the relaxation process and the neural network technique. Some advantages of using neural networks are that its natural character of parallel computation enables neural networks could solve complicated problems efficiently and the structure of neural networks can be implemented by VLSI technology at a more reasonable cost. So the relaxation process based on the neural network technique can be accomplished in real time. But many investigations which used neural networks to relaxation labeling are based on the standard Hopfield neural network [1] and [11]. The SHNN contains penalty parameters, thus it only generates the approximate solution and has an implementation problem when penalty parameters are very large. In this paper, an improved Hopfield neural network presented in [3] has been utilized to fulfill the relaxation process. The IHNN model uses fewer neurons than that of SHNN. The activation function of IHNN has a "saturation" nonlinearity form which is easier to be implemented than the sigmoid function used in SHNN. Moreover, the IHNN approach is demonstrated by variational inequality technique to have the ability of obtaining an exact optimal solution [3]. According to experimental results, the IHNN approach can obtain a better labeling performance than that of SHNN.

The remainder of this paper is organized as follows. Section 2 provides the dynamical differential equations and architectures of the improved Hopfield neural network. Experimental results are shown and discussed in Section 3. Section 4 concludes this paper with final remarks.

## 2 Improved Hopfield Neural Network Model

First, the constrained optimization problem which maximizes the average local consistency function (4) is stated as follows

$$\max Z(p) = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{h=1}^{n}\sum_{k=1}^{m} p_{ij} r(i,j;h,k) p_{hk}, \tag{5}$$

$$\text{s.t.} \ \sum_{j=1}^{m} p_{ij} = 1,$$

$$0 \le p_{ij} \le 1,\ i = 1, 2, \cdots, n; \quad j = 1, 2, \cdots, m.$$

This is a quadratic optimization problem with equality and inequality constraints. In the neural network literature, there exist a few NN models for solving quadratic optimization problems. Some of them are based on the penalty function method [2] and [8]. The fundamental idea behind the penalty function method is that, when a constraint violation occurs, the magnitude and direction of the violation are fed back to adjust the states of neurons. But penalty parameters cannot be infinite, the network only generates the approximate solution. Wang et al. presented several Hopfield-type neural network approaches (Lagrangian neural network, primal-dual neural network, primal neural network and dual neural network) which fulfill the Karush-Kuhn-Tucker (KKT) condition to obtain an exact optimal solution [3], [9] and [10]. According to [9] and

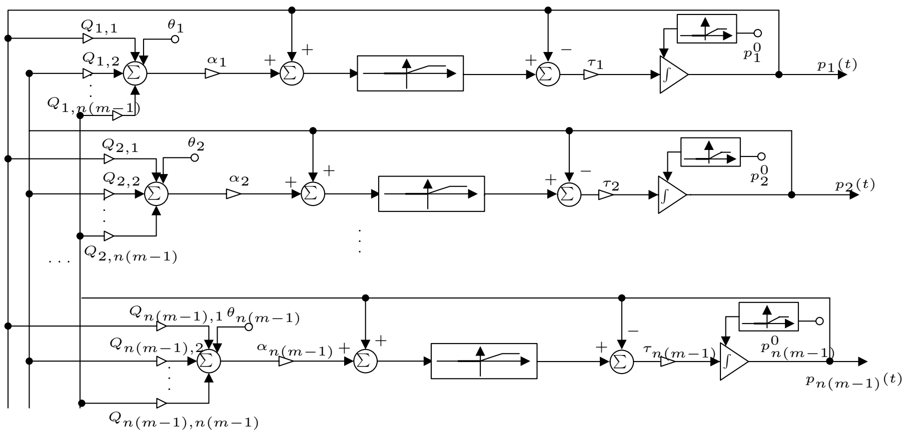

**Fig. 1.** Functional block diagram for IHNN model

[10], the primal-dual neural network and the dual neural network usually require the objective function convexity, this requirement can not be guaranteed by the quadratic optimization problem defined by (5). An improved continuous-time Hopfield neural network presented in [3] is adopted in this paper. This network model can deal with bound-constrained nonlinear optimization with any continuously differentiable objective function which is not necessarily quadratic or convex. The quadratic optimization problem considered by IHNN in [3] is described as follows

$$\begin{aligned} &\min E(x) = \frac{1}{2}x^T Q x + x^T \theta, \\ &\text{s.t. } a \le x \le b \end{aligned} \tag{6}$$

where $x = (x_1, x_2, \cdots, x_n)^T \in \Re^n$ is a decision vector; $a = (a_1, a_2, \cdots, a_n)^T \in \Re^n$ and $b = (b_1, b_2, \cdots, b_n)^T \in \Re^n$ are respectively given constant lower and upper bounds with $a_i \le b_i$ $(i = 1, 2, \cdots, n)$; $Q \in \Re^{n\times n}$ is a symmetric weighted matrix; $\theta = (\theta_1, \theta_2, \cdots, \theta_n)^T \in \Re^n$ is a constant vector and the superscript $T$ denotes the transpose operator.

Note that the quadratic problem defined by (5) has equality constraints. In order to utilize the IHNN approach, equality constraints have to be canceled by replacing $p_{ij_0}$ $(\forall j_0 \in \{1, 2, \cdots, m\})$ in (5) with $1 - \sum_{j=1, j\ne j_0}^m p_{ij}$, then the objective function of the optimization problem defined by (5) can be reformulated as follows

$$\begin{aligned} Z(p) &= \sum_{i=1}^n \sum_{j=1}^n \sum_{h=1}^n \sum_{k=1}^m p_{ij} r(i,j;h,k) p_{hk} \\ &= \sum_{i=1}^n \sum_{j=1\, j\ne j_0}^m \sum_{h=1}^n \sum_{k=1, k\ne j_0}^m p_{ij} p_{hk} Q_{ij,hk} + \\ &\quad \sum_{i=1}^n \sum_{j=1, j\ne j_0}^m p_{ij}\theta_{ij} + \sum_{i=1}^n \sum_{h=1}^n r(i,j_0;h,j_0) \\ &= p^T Q p + p^T \theta + C. \end{aligned} \tag{7}$$

where $p = \left(p_{11}, \cdots, p_{1(j_0-1)}, p_{1(j_0+1)}, \cdots, p_{1m}, \cdots, p_{n1}, \cdots, p_{n(j_0-1)}, p_{n(j_0+1)}, \cdots, p_{nm}\right)^T \in \Re^{n(m-1)}$ is the decision variables; $Q_{ij,hk} = r(i,j;h,k) - r(i,j;h,j_0) - r(i,j_0;h,k) + r(i,j_0;h,j_0)$, $Q = (Q_{ij,hk}) \in \Re^{n(m-1)\times n(m-1)}$ is the symmetric weighted matrix; $\theta_{ij} = \sum_{h=1}^{n} r(i,j;h,j_0) + \sum_{h=1}^{n} r(h,j_0;i,j) - \sum_{h=1}^{n} r(h,j_0;i,j_0) - \sum_{h=1}^{n} r(i,j_0;h,j_0)$, $\theta = (\theta_{ij}) \in \Re^{n(m-1)}$ and $C = \sum_{i=1}^{n}\sum_{h=1}^{n} r(i,j_0;h,j_0)$.

Thus the problem defined by (5) is converted to the following quadratic optimization problem with only bound constraints.

$$\begin{aligned} &\min Z(p) = -(p^TQp + p^T\theta + C), \\ &\text{s.t. } \mathbf{0} \le p \le \mathbf{1}, \end{aligned} \tag{8}$$

where $\mathbf{0} = (0,0,\cdots,0)^T \in \Re^{n(m-1)}$ and $\mathbf{1} = (1,1,\cdots,1)^T \in \Re^{n(m-1)}$.

According to [3], an improved Hopfield neural network model can be designed to resolve above problem. The differential equations which determine the IHNN configuration are described as follows

$$\Gamma \frac{dp}{dt} = -p + g\left(p + \Lambda(2Qp + \theta)\right) \tag{9}$$

where $\Gamma = \operatorname{diag}\left(\tau_1, \tau_2, \cdots, \tau_{n(m-1)}\right)$ and $\Lambda = \operatorname{diag}\left(\alpha_1, \alpha_2, \cdots, \alpha_{n(m-1)}\right)$ are time constant positive defined matrix and scaling positive defined matrix, respectively. These parameters can be used to control the convergence rate of the IHNN model. The activation function $g(x) = \left(g_1(x), g_2(x), \cdots, g_{n(m-1)}(x)\right)^T$ and $g_i(x)$ is defined by following equation

$$g_i(x) = \begin{cases} 0, & \text{if } x < 0; \\ 1, & \text{if } x > 1; \\ x, & \text{if } 0 \le x \le 1. \end{cases}$$

$g(x)$ can be regarded as the projection operator of $\Re^{n(m-1)}$ to the closed convex feasible region $\Omega = [0,1] \times [0,1] \cdots \times [0,1]$.

The IHNN model has been demonstrated to have following features by variational inequality and ordinary differential equation techniques in [3]. 1) The IHNN model is *regular* in the sense that any optimal solution to problem defined by (8) is also an equilibrium point of IHNN. Moreover, if the objective function is convex, the set of equilibrium point of IHNN is equal to the set of the optimum of optimization problem defined by (8). 2) The IHNN model is *quasiconvergent* in the sense that the trajectory of IHNN cannot escape from the feasible region and will converge to the set of equilibrium point of IHNN for any network initial state in the feasible region. This guarantees the IHNN model is stable. 3) If the initial state of IHNN is in the feasible region, the value of objective function will decrease along the corresponding solution trajectory. Thus, although the objective function of the optimization problem defined by (8) is not convex, which means that the equilibrium point of IHNN may not be the optimum of optimization problem defined by (8), the convergence point of IHNN can still obtain a better performance than the network initial state. Also because of the non-convexity of the objective function, there are maybe more than one equilibrium point of IHNN. In order to make IHNN converge to the equilibrium point resulted in a better labeling performance, the initial state of IHNN cannot be selected randomly in the feasible region. The initial state which is computed by analyzing the corresponding relaxation labeling problem can be regarded as a prior knowledge, and the solution trajectory of IHNN is expected to converge to the equilibrium point which is nearest to the initial state. After IHNN converges, $p_{ij_0}$ can be obtained by the relationship $p_{ij_0} = 1 - \sum_{j=1, j\neq j_0}^{m} p_{ij}$. Then select the maximum value $p_{ij_m}$ among $p_{i1}, p_{i2}, \cdots, p_{im}$. Object $i$ is assigned to label $j_m$.

The dynamical equations (9) of IHNN indicate that the network is composed by only one layer of $n(m-1)$ neurons which is fewer than that of SHNN which is composed of $nm$ neurons [1]. The IHNN has only one nonlinearity form "saturation" which is easier to be implemented by an operational amplifier than the sigmoid function used in SHNN. The IHNN approach is not based on the penalty function method, it yields an exact optimal solution while SHNN contains finite penalty parameters which can only generate an approximate optimal solution.

The block diagram of the IHNN controller is shown in Fig. 1. The weighted connections of network and the initial network state have to be determined according to the corresponding relaxation labeling problem.

## 3 Experimental Results

To demonstrate the efficiency of the IHNN approach, a typical image processing problem, thresholding, is performed. In other word, to determine each pixel of a noise corrupted image belong to either object or background. The original gray image is a Chinese character "cheng" shown in Fig. 2. The size of this image is $40 \times 40$. The most of intensity values of character is 20 while the intensity value of background is 255. Then this image is corrupted by a zero means white noise. The corrupted image is depicted in Fig. 3. It is clear that thresholding by simply using a single thresholding value for the entire image does not work, so relaxation method is used to deal with this problem.

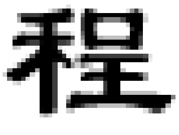

**Fig. 2.** The Chinese character "cheng" without noise corruption

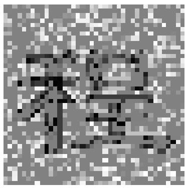

**Fig. 3.** The Chinese character "cheng" with white noise corruption

The object of this relaxation labeling problem is each pixel point of the corrupted image and two kinds of labels, character and background, will be assigned to each object. For the pixel point $A_{x,y}$, $p_{x,y,1}$ denotes the probability of this point having character label and $p_{x,y,0} = 1 - p_{x,y,1}$ denotes the case of background label. Then the initial probability of each pixel and compatibility coefficients have to be determined. For the initial probability, a scheme to determine the likelihood of the pixel being the character or background is needed. According to the analysis in Section 2, a good initial probability assignment results in a good labeling performance. So the likelihood determination scheme is very important to the problem resolution. Here a simple method presented in [4] is adopted. Specifically, let $V_{x,y}$ be the gray level of pixel point $A_{x,y}$, $V_{max}$ and $V_{min}$ are the maximum and minimum gray levels of the entire image, respectively. Then initial probability of pixel point $A_{x,y}$ can be obtained by following equations

$$p^0_{x,y,0} = \frac{V_{x,y}}{V_{max} - V_{min}}, \tag{10}$$

$$p^0_{x,y,1} = 1 - \frac{V_{x,y}}{V_{max} - V_{min}}. \tag{11}$$

For compatibility coefficients, suppose that each pixel point $A_{x,y}$ only correlates to its nearest 8-neighborhood points. So the compatibility coefficient for any point not in the 8-neighborhood will be zero. For each point $A_{x+i,y+j}$ $(i, j \in \{-1, 0, 1\})$, there are four compatibility coefficients, $r(x, y, 1; x + i, y + j, 1)$, $r(x, y, 1; x+i, y+j, 0)$, $r(x, y, 0; x+i, y+j, 1)$ and $r(x, y, 0; x+i, y+j, 0)$. A general method (12) proposed in [6], which is based on the mutual information of labels at neighboring pixels, is utilized to compute these compatibility coefficients. The computation result is shown in Table 1.

$$r(x, y, \lambda; x + i, y + j, \tilde{\lambda}) = \log \frac{n \sum\limits_{(x,y)} p_{x,y,\lambda} p_{x+i,y+j,\tilde{\lambda}}}{\sum\limits_{(x,y)} p_{x,y,\lambda} \sum\limits_{(x,y)} p_{x,y,\tilde{\lambda}}} \tag{12}$$

where $\lambda, \tilde{\lambda} \in \{0, 1\}$.

**Table 1.** Compatibility coefficients computed by (12) where each element in row $(\lambda, \tilde{\lambda})$ and column($(x, y)$,$(x + i, y + j)$) means the coefficient $r(x, y, \lambda; x + i, y + j, \tilde{\lambda})$

| $(\lambda, \tilde{\lambda})$ | $(x, y)$ $(x-1, y-1)$ | $(x, y)$ $(x-1, y)$ | $(x, y)$ $(x-1, y+1)$ | $(x, y)$ $(x, y+1)$ |
|---|---|---|---|---|
| 0; 0 | 0.023070 | 0.027393 | 0.025787 | 0.040600 |
| 0; 1 | -0.029055 | -0.034715 | -0.032637 | -0.051198 |
| 1; 0 | -0.036887 | -0.038016 | -0.032475 | -0.049311 |
| 1; 1 | 0.044478 | 0.045800 | 0.039289 | 0.058908 |
|  | $(x, y)$ $(x+1, y+1)$ | $(x, y)$ $(x+1, y)$ | $(x, y)$ $(x+1, y-1)$ | $(x, y)$ $(x, y-1)$ |
| 0; 0 | 0.029292 | 0.029778 | 0.024255 | 0.037603 |
| 0; 1 | -0.037177 | -0.037809 | -0.030661 | -0.048042 |
| 1; 0 | -0.032316 | -0.036012 | -0.034328 | -0.053104 |
| 1; 1 | 0.039101 | 0.043452 | 0.041473 | 0.063259 |

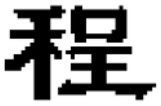

**Fig. 4.** The thresholding image processed by the IHNN approach

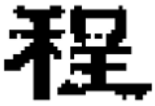

**Fig. 5.** The thresholding image processed by the SHNN approach

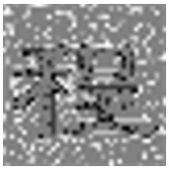

**Fig. 6.** The Chinese character "cheng" with color noise corruption

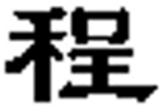

**Fig. 7.** The thresholding result of figure 6 by the IHNN approach

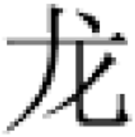

**Fig. 8.** The Chinese character "long" without noise

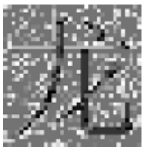

**Fig. 9.** The Chinese character "long" with white noise corruption

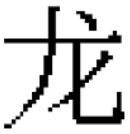

**Fig. 10.** The thresholding image processed by the IHNN approach

Now, the IHNN model is simulated on an *IBM* personal computer. The differential equations (9) which determine the IHNN configuration are solved by Matlab ode45 method and the time constants matrix $\Gamma$ and scaling matrix $\Lambda$ in (9) are set to be $100 \times I$ ($I$ is a unity matrix) and $100 \times I$, respectively.

The thresholding image processed by the IHNN approach is shown in Fig. 4, which is almost same with the original image. Then the SHNN approach is used to process the corrupted image again in order to compare with the IHNN method. The gain $\lambda$ of SHNN is set to be 0.1 and the initial probability and compatibility coefficients are computed by the method used in [1]. The thresholding image processed by the SHNN approach is shown in Fig. 5. According to the experimental result, the IHNN method can obtain a better performance than that of SHNN. In order to verify the robustness of IHNN algorithm, some other type of noise is generated to corrupt the original image. This color noise is the output signal of the filter $\frac{1}{1-z^{-1}}$ whose input signal is the original white noise. The color noise corrupted image is shown in Fig. 6, and the thresholding image is depicted in Fig. 7. It is obvious that the IHNN algorithm can also obtain a good thresholding result.

Another thresholding experiment is to deal with a Chinese character "long" with a different font type. The original image is shown in Fig. 8 and the noise corrupted image is shown in Fig. 9. The initial probability values and compatibility coefficients are computed by the same method used in the first experiment. $\Gamma$ and $\Lambda$ of the IHNN model are set to be $100 \times I$ and $100 \times I$. According to the experimental result shown in Fig. 10, the IHNN approach can still obtain a good labeling performance.

## 4 Concluding Remarks

The relaxation labeling is a useful technique to deal with local ambiguity and achieve consistency. In [1], some useful comments indicate that several common properties exist in the relaxation process and the neural network technique. Recently, many investigations in this field are based on the standard Hopfield neural network. In this paper, an improved Hopfield neural network model presented in [3] is utilized to fulfill the relaxation labeling process. Compared to the SHNN, this IHNN approach has some advantages that 1) The IHNN uses fewer neurons than that of SHNN. 2) The activation function of IHNN has a "saturation" nonlinearity form which is easier to be implemented than that of SHNN. 3) The IHNN is not based on the penalty function method and can generate an exact optimal solution while SHNN contains the finite penalty parameters which result in an implementation problem when penalty parameters are very large. According to the experimental results, the IHNN method can obtain a better labeling performance than that of SHNN. Because the neural network approach can be used as a guidance to design analog or digital circuits, thus any real-time relaxation application can be accomplished by the IHNN method. Note that the relaxation labeling problem is essentially a non-convex quadratic optimization problem. The IHNN model used here is only *regular* but not *complete* (*complete* means the set of equilibrium point of the neural network model is equal to the set of the optimum of quadratic optimization problem). So some advanced *complete* neural network model can be further investigated. Moreover, how to select the network initial state appropriately according to the corresponding relaxation problem is also an interesting topic for future research.

## Acknowledgments

This research was supported in part by the National Natural Science Foundation of China (Grants 60205004, 50475179 and 60334020), the National Basic Research Program (973) of China (Grant 2002CB312200), and the Hi-Tech R&D Program (863) of China (Grants 2002AA423160 and 2005AA420040).

## References

1. Yu, S. S., Tsai, W. H.: Relaxation by the Hopefield Neural Network. Pattern Recongition. **25** (1992) 197-209
2. Tank, D. W., Hopfield, J. J.: Simple Neural Optimization Networks: an A/D Converter, Signal Decision Circuit and a Linear Programming Circuit. IEEE Transcations on Circuits and Systems. **35** (1988) 554-562
3. Liang, X. B., Wang, J.: A Recurrent Neural Network for Nonlinear Optimization with a Continuously Differentiable Objective Function and Bound Constraints. IEEE Transcations on Neural Networks. **11** (2000) 1251-1262
4. Rosenfeld, A., Kak, A. C.: Digital Picture Processing Volume 2. New York: Academic Press, (1982)
5. Rosenfeld, A., Hummel, R. A., Zucker, S. W.: Scene Labeling by Relaxation Operations. IEEE Transcations on System, Man and Cybernetics. **6** (1976) 420-433
6. Peleg, S., Rosenfeld, A.: Determining Compatibility Coefficients for Curve Enchancement Relaxation Processes. IEEE Transcations on System, Man and Cybernetics. **8** (1978) 548-555
7. Hummel, R. A., Zucker, S. W.: On the Foundations of Relaxation Labeling. IEEE Transcations on Pattern Analysis and Machine Intelligence. **5** (1983) 267-287
8. Kennedy, M. P., Chua, L. O.: Neural Networks for Nonlinear Programming. IEEE Transcations on Circuits and Systems Part A. **35** (1988) 554-562
9. Zhang, Y. N., Wang, J., Xia, Y. S.: A Dual Neural Network for Redundancy Resolution of Kinematically Redundant Manipulators Subject to Joint Limits and Joint Velocity Limits. IEEE Transcations on Neural Networks. **14** (2003) 658-667
10. Xia, Y. S., Wang, J.: A Recurrent Reural Network for Solving Nonlinear Convex Programs Subject to Linear Constraints. IEEE Transcations on Neural Networks. **16** (2005) 379-386
11. Sang, N., Zhang, T. X.: Segmentation of FLIR Images by Hopfield Neural Network with Edge Constraint. Pattern Recongition. **34** (2001) 811-821
12. Kurugollu, F., Sankur, B., Harmanci, A. E.: Image Segmentation by Relaxation Using Constraint Satisfaction Neural Network. Image and Vision Computing. **20** (2002) 483-497