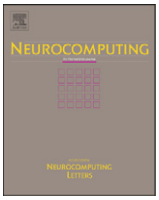
Neurocomputing 71 (2008) 3264–3274

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Hough transform neural network for pattern detection and seismic applications

Kou-Yuan Huang *, Kai-Ju Chen, Jiun-Der You, An-Ching Tung

Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan

ARTICLE INFO

Available online 25 June 2008

Keywords:
Hough transform neural network
Seismic pattern detection
Reflection wave
Hyperbolic pattern detection

ABSTRACT

Hough transform neural network is adopted to detect the line pattern of direct wave and the hyperbolic pattern of reflection wave in a one-shot seismogram. We use time difference from point to hyperbola and line as the distance in the pattern detection of seismic direct and reflection waves. This distance calculation makes the parameter learning feasible. One set of parameters represents one pattern. Many sets of parameters represent many patterns. The neural network can calculate the distances from point to many patterns as total error. The parameter learning rule is derived by gradient descent method to minimize the total error. The network is applied to three kinds of data in the experiments. One is the line and hyperbolic pattern in the image data. The second is the simulated one-shot seismic data. And the last is the real one-shot seismic data. Experimental results show that lines and hyperbolas can be detected correctly in three kinds of data. The method can also tolerate certain level of noise data. The detection results in the one-shot seismogram can improve the seismic interpretation and further seismic data processing.

1. Introduction

Hough transform (HT) was used to detect the parameterized shapes by mapping original image data in the image space into the parameter space [5,6,10,12]. The purpose of HT was to find the peak value (maximum) in the parameter space. The coordinates of a peak value in parameter space were corresponding to a shape in the image space. Calculating such transformation was time consuming, the memory in parameter space was too large, and the peak determination was not easy in the parameter space.

Neural network was developed to solve the HT problem [1–3]. The Hough transform neural network (HTNN) was designed for detecting lines, circles, and ellipses [1–3]. The determination of parameters of objects was by the neural network learning, not by the mapping to the parameter space. But there was no application to the detection of hyperbola.

Seismic pattern recognition plays an important role in oil exploration. In a one-shot seismogram, the travel-time curve of direct wave pattern is a straight line and the reflection wave pattern is a hyperbola. In 1985 and 1987, Huang et al. had applied the HT to detect the line pattern of direct wave and the hyperbolic pattern of reflection wave in a one-shot seismogram [7,8]. However, it had the same serious drawbacks in the calculation as the conventional HT.

In 2006, Huang et al. [9] applied HTNN to detect the line pattern of direct wave and the hyperbolic pattern of reflection wave in a one-shot seismogram. However, the distance from point to line and hyperbola using different distance definitions, determination of two-stage learning and decreasing basis in Gaussian basis function make the learning complex. Here, we also take the advantage of HTNN. We define the vertical time difference as the distance from point to line and to hyperbola that makes the definition consistence. Moreover, the learning process is simplified to one-stage and the preprocessing of the envelope processing, threshold processing, and peak detection processing is taken to improve the detection result. We also have the real seismic data experiment.

2. Seismic signals

Seismic exploration is an essential procedure in oil and gas exploration. Seismic signal is produced by explosion and receiving. After explosion, there are direct waves along the ground surface and reflection waves from the reflection layer.

2.1. Direct wave

There are two basic kinds of seismic waves: the direct wave and the reflection wave [4,13,14]. Direct wave propagates along the surface to the receivers. Fig. 1 shows the wave path of the

* Corresponding author.
E-mail address: kyhuang@cs.nctu.edu.tw (K.-Y. Huang).

0925-2312/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2008.04.034

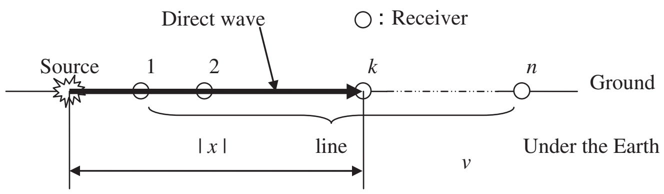


Fig. 1. Wave path of the direct wave.

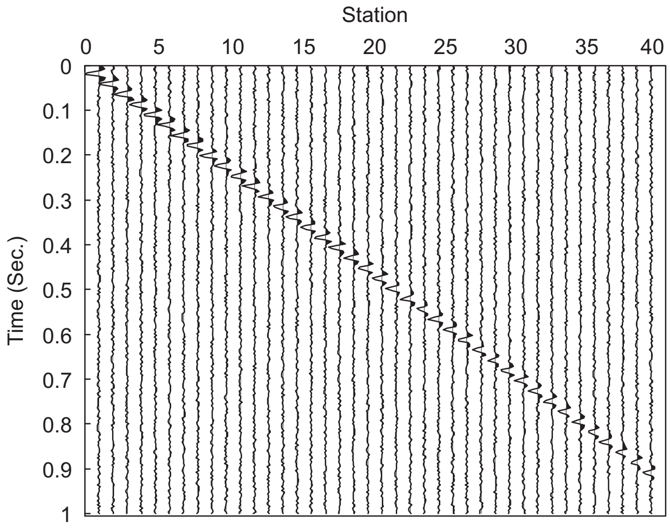


Fig. 2. Direct wave in the one-shot seismogram.

direct wave, and the time when the direct wave reaches the receiver is in Eq. (1), where x is the distance from the source to the receiver. Fig. 2 is the direct wave in the one-shot seismogram with 40 traces. The time–distance curve is a line. Wave velocity in the layer is 2200 m/s, and the distance of the receiving station is 50 m. The sampling interval is 0.004 s:

$$t = \frac{x}{v}. \tag{1}$$

2.2. Reflection wave

The reflection wave propagates through the medium and is reflected back from the reflection layer. There are two cases: from the horizontal reflection layer and from the dipping reflection layer. The first case is the horizontal reflection layer as shown in Fig. 3. The time of the wave reaching the kth receiver is derived in Eq. (2), and the time–distance curve is a hyperbola as shown in Fig. 4:

$$t = \frac{(\overline{OQ}+\overline{QP})}{v} = \frac{\overline{O'P}}{v} = \frac{\sqrt{\overline{OO'}^2+\overline{OP}^2}}{v} = \frac{\sqrt{(2d)^2+x^2}}{v}. \tag{2}$$

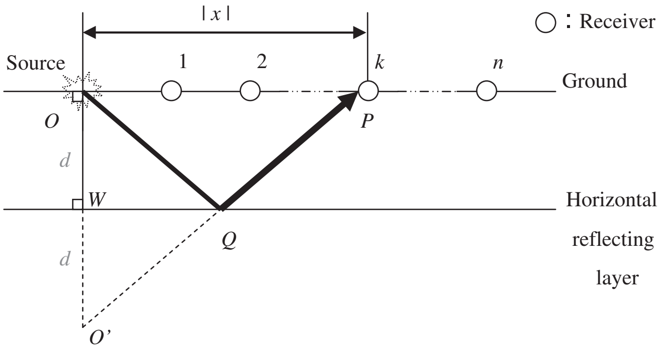


Fig. 3. Reflection from the horizontal reflection layer.

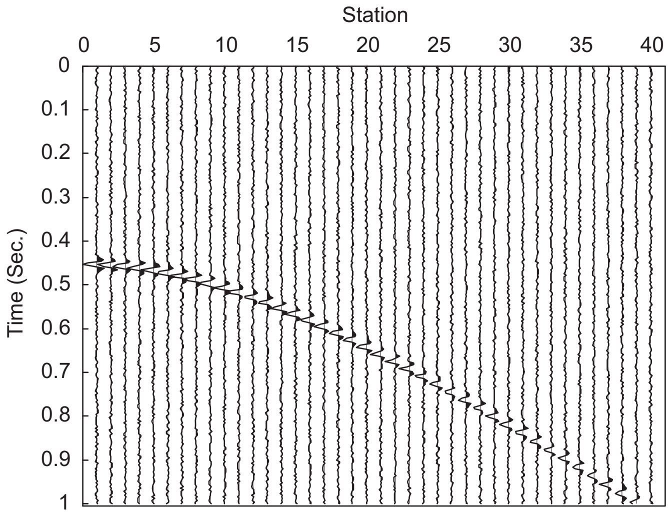


Fig. 4. One-shot seismogram: reflection wave from the horizontal reflection layer.

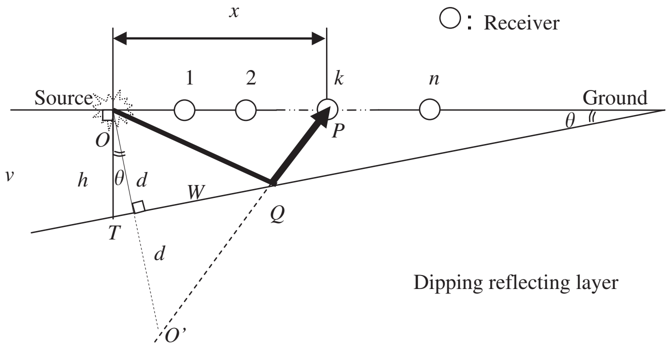


Fig. 5. Reflection from the dipping reflection layer.

The other case is a dipping reflection layer as shown in Fig. 5. In this case, we use trigonometric laws of cosine to derive the relation between receiving time t and the distance x in Eq. (3). After further algebraic calculation, we find it is also a hyperbola (Eq. (4)). Fig. 6 shows the one-shot seismogram of the reflection wave from the dipping reflection layer. Note here, the wave from dipping reflection layer shifts right related to the wave which is from horizontal reflection layer:

$$\begin{aligned} t &= \frac{(\overline{OQ}+\overline{QP})}{v} = \frac{\overline{O'P}}{v} \\ &= \frac{\sqrt{\overline{O'O}^2+\overline{OP}^2-2\overline{O'O}\,\overline{OP}\cos(\text{angle}(\angle O'OP))}}{v} \\ &= \frac{\sqrt{\overline{O'O}^2+\overline{OP}^2-2\overline{O'O}\,\overline{OP}\cos(90^\circ-\theta)}}{v} \\ &= \frac{\sqrt{(2d)^2+x^2-4dx\sin\theta}}{v}, \end{aligned} \tag{3}$$

$$t = \frac{\sqrt{(2d)^2 + (x - 2d\sin\theta)^2 - (2d)^2\sin^2\theta}}{v} = \frac{\sqrt{(2d\cos\theta)^2 + (x - 2d\sin\theta)^2}}{v}. \qquad (4)$$

3. Seismic pattern detection system

To detect parameters of the line (direct wave) and hyperbola (reflection wave), we adopt the HTNN. The proposed seismic pattern detection system is shown in Fig. 7.

3.1. Preprocessing

The input seismogram including the direct wave and reflection wave in Fig. 8 passes through envelope processing, threshold processing, and peak detection processing [7]. For seismic data, $s(x_i, t_i)$, $1 \leqslant x_i \leqslant 65$, $1 \leqslant t_i \leqslant 512$, after envelope processing, $s(x_i, t_i)$ becomes $s'(x_i, t_i)$, we set a threshold T. Through threshold processing, $s'(x_i, t_i) \geqslant T$, and peak detection processing, data become the object points, $\mathbf{x_i} = [x_i, t_i]^{\mathrm{T}}$, $i = 1, 2, \ldots, n$. Fig. 9 is the result of preprocessing. After preprocessing, the object points enter the HTNN.

3.2. Hough transform neural network (HTNN)

The adopted HTNN consists of three layers: distance layer, radial basis function (RBF) layer, and the total error layer. The network is shown in Fig. 10. It is an unsupervised network capable of detecting m parameterized objects: lines and hyperbolas, simultaneously. The objects can fit the data. And the error must be minimized. Gradient descent method is used in the parameter learning.

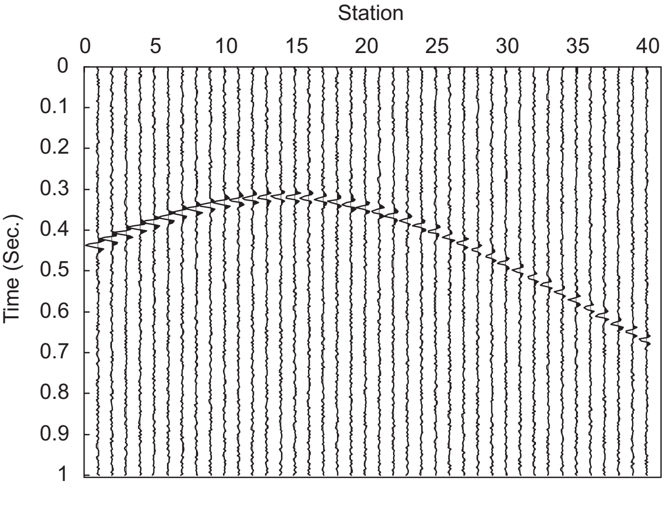


Fig. 6. One-shot seismogram: reflection wave from the dipping reflection layer.

In Fig. 10, input vector $\mathbf{x}_i = [x_i, t_i]^{\mathrm{T}}$ is the ith point of the image, where $i = 1, 2, \ldots, n$. In the preprocessed seismic image, x_i is the trace index between shot point and receiving station, and t_i is index in time coordinate. Each pattern (line or hyperbola) has one set of parameters. The number of patterns, m, must be given before HTNN learning. Input each point $\mathbf{x}_i$ into distance layer, we calculate the distance $d_{ik} = D_k(\mathbf{x}_i) = D_k(x_i, t_i)$ from $\mathbf{x}_i$ to the kth pattern (line or hyperbola), $k = 1, 2, \ldots, O_{\mathrm{L}}$ or O_{H}, where O_{L} is the number of line patterns and O_{H} is the number of hyperbolic patterns. Then, d_{ik} passes through the RBF layer and the activation output is $e_{ik} = 1 - f(d_{ik})$, where $f(\cdot)$ is a Gaussian basis function, i.e., $f(d_{ik}) = \exp(-d_{ik}^2 2/\sigma^2)$, and e_{ik} is the error or the modified distance from the ith point $\mathbf{x}_i$ to the kth pattern. Thus, when d_{ik} is near zero, e_{ik} is also near zero. In the last layer, we calculate the total error from $\mathbf{x}_i$ to all m patterns, $E_i = e_{i1} \cdot e_{i2} \cdot \cdots \cdot e_{im}$. When $\mathbf{x}_i$ belongs to one pattern, then $e_{ik} = 0$, and $E_i = 0$.

3.3. Distance layer

3.3.1. Distance from point to hyperbola

In a one-shot seismogram, the reflection wave pattern is a hyperbola. Note that, for seismic pattern, there is no rotated hyperbola. So the equation of the kth hyperbola is

$$-\left(\frac{x - x_{0,k}}{a_k}\right)^2 + \left(\frac{t - t_{0,k}}{b_k}\right)^2 - 1 = 0. \qquad (5)$$

We consider the positive x side, the equation becomes:

$$H_k(\mathbf{x}) = b_k\sqrt{\left(\frac{x - x_{0,k}}{a_k}\right)^2 + 1} - (t - t_{0,k}) = 0. \qquad (6)$$

We minimize the distance from point $\mathbf{x_i}$ to the hyperbola. That is

$$\text{minimize } \tfrac{1}{2}||\mathbf{x} - \mathbf{x}_i||^2 \text{ subject to } H_k(\mathbf{x}) = 0. \qquad (7)$$

From Lagrange method, the Lagrange function is

$$l(\mathbf{x}, \lambda) = \frac{1}{2}||\mathbf{x} - \mathbf{x}_i||^2 + \lambda\left[b_k\sqrt{\left(\frac{x - x_{0,k}}{a_k}\right)^2 + 1} - (t - t_{0,k})\right]. \qquad (8)$$

By the first-order necessary condition, we have the following equations:

$$\frac{\partial}{\partial x} l(\mathbf{x}, \lambda) = (x - x_i) + \lambda\frac{b_k}{a_k^2}(x - x_{0,k}) \Big/ \sqrt{(x - x_{0,k}/a_k)^2 + 1} = 0,$$

$$\frac{\partial}{\partial t} l(\mathbf{x}, \lambda) = (t - t_i) - \lambda = 0,$$

$$\frac{\partial}{\partial \lambda} l(\mathbf{x}, \lambda) = b_k\sqrt{\left(\frac{x - x_{0,k}}{a_k}\right)^2 + 1} - (t - t_{0,k}) = 0. \qquad (9)$$

To solve the equations in Eq. (9), we must take fourth-order complex computations. So alternatively we consider the time difference as the distance from a point $\mathbf{x}_i = (x_i, t_i)$ to a hyperbola in seismic pattern detection case. The distance is defined as

$$d_{ik} = t(x_i) - t_i = b_k\sqrt{\left(\frac{x_i - x_{0,k}}{a_k}\right)^2 + 1} - (t_i - t_{0,k}) = H_k(\mathbf{x}_i). \qquad (10)$$

Fig. 11 illustrates the distance from point to hyperbola in the difference of vertical time direction.

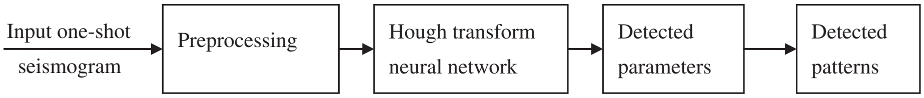


Fig. 7. System of seismic pattern detection.

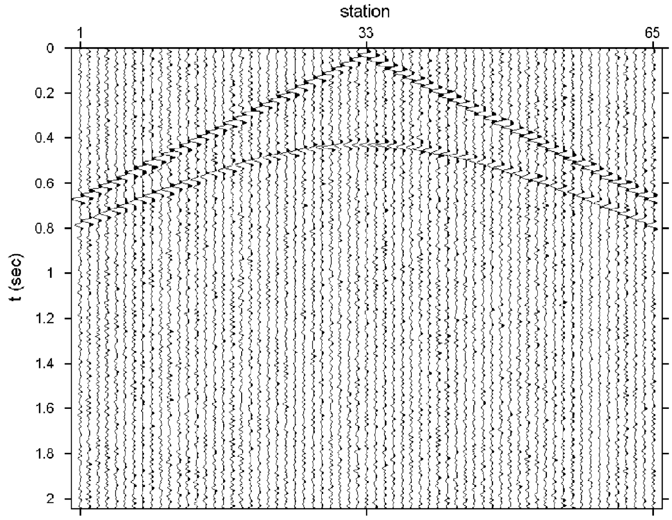


Fig. 8. One-shot seismogram.

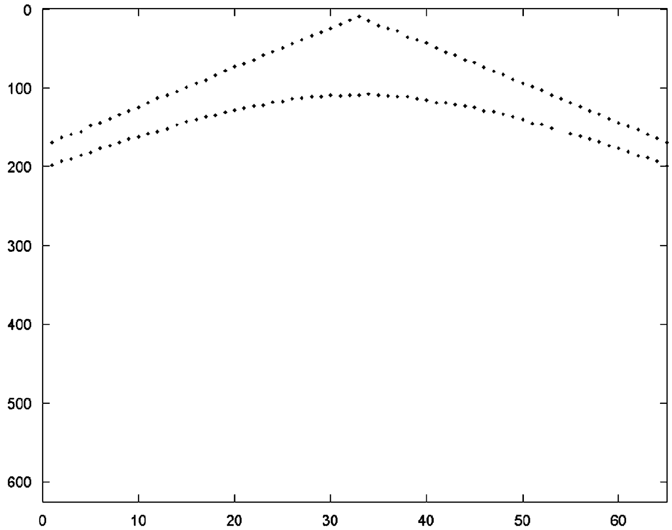


Fig. 9. Result of preprocessing.

3.3.2. *Distance from point to line*

Although Basak and Das proposed HTNN to detect lines, they used the second-order equation of conoidal shapes [2]. Here, we use the slope–intercept form of straight-line equation in the analysis.

The equation of the kth line is

$$t = m_k x + b_k, \tag{11}$$

or in the form:

$$L_k(\mathbf{x}) = m_k x - t + b_k = 0. \tag{12}$$

The distance from the point $\mathbf{x}_i$ to the kth line is a constraint minimization problem:

$$\text{minimize } \tfrac{1}{2}||\mathbf{x} - \mathbf{x}_i||^2 \text{ subject to } L_k(\mathbf{x}) = 0. \tag{13}$$

However, for the case of seismic pattern, to have the same distance measure, the distance from a point to the line is also considered as time difference from the point $\mathbf{x}_i = (x_i, t_i)$ to the line.

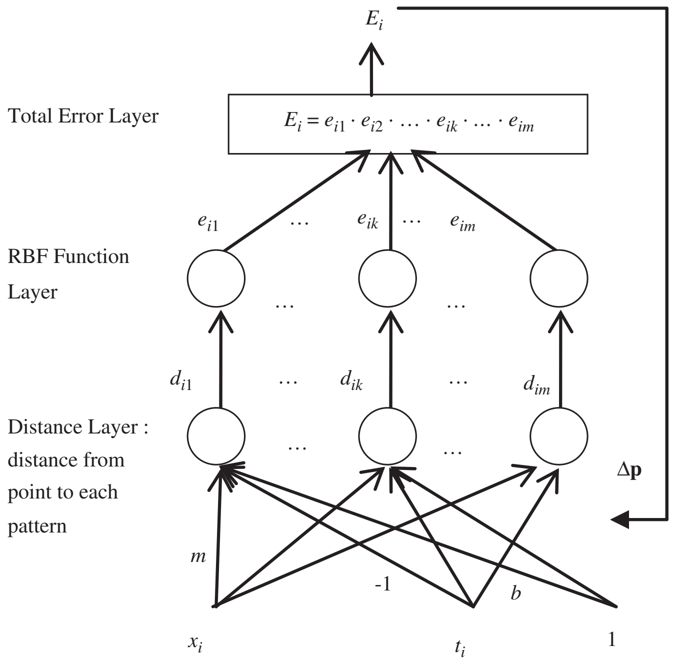

E_i
Total Error Layer
$E_i = e_{i1} \cdot e_{i2} \cdot \ldots \cdot e_{ik} \cdot \ldots \cdot e_{im}$
e_{i1}
e_{ik}
e_{im}
RBF Function Layer
d_{i1}
d_{ik}
d_{im}
Distance Layer : distance from point to each pattern
$\Delta\mathbf{p}$
m
-1
b
x_i
t_i
1

Fig. 10. Hough transform neural network.

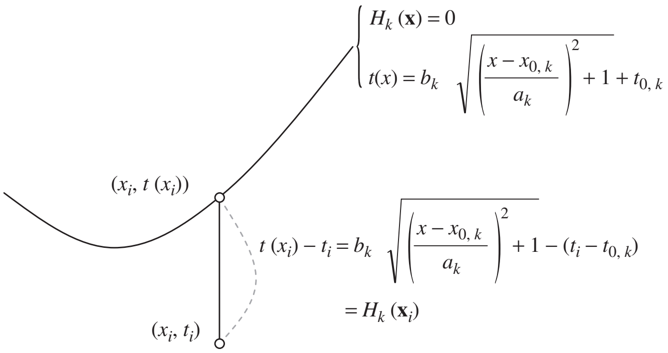

$H_k(\mathbf{x}) = 0$
$t(x) = b_k \sqrt{\left(\frac{x - x_{0,k}}{a_k}\right)^2 + 1} + t_{0,k}$
$(x_i, t(x_i))$
$t(x_i) - t_i = b_k \sqrt{\left(\frac{x - x_{0,k}}{a_k}\right)^2 + 1} - (t_i - t_{0,k})$
$= H_k(\mathbf{x}_i)$
(x_i, t_i)

Fig. 11. Distance from point to hyperbola.

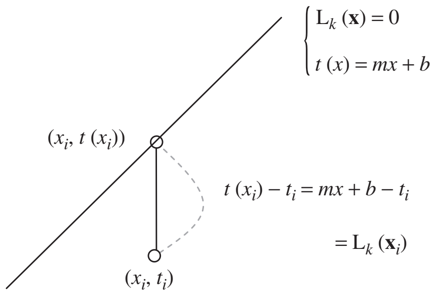


Fig. 12. Distance from point to line.

The distance used here is

$$d_{ik} = t(x_i) - t_i = m_k x - t_i + b_k = L_k(\mathbf{x}_i). \tag{14}$$

Fig. 12 illustrates the distance from point to line in the difference of vertical time direction.

3.4. RBF function layer

After the distance calculation from a point to a pattern, the modified distance or the error is defined as

$$e_{ik} = 1 - f(d_{ik}) = 1 - \exp\left(-\frac{d_{ik}^2}{\sigma^2}\right), \tag{15}$$

where $f(d_{ik})$ is the RBF. The σ in Gaussian basis function controls the effected range. For the larger σ, the effected range is larger. If we do not use RBF, we can use $f(d_{ik}) = |d_{ik}|$.

3.5. Total error layer

We consider that the multiplication of m errors from RBF function layer is the total error. So the total error for point $\mathbf{x}_i$ is defined as

$$E_i = C(e_{i1}, \ldots, e_{ik}, \ldots, e_{im}) = e_{i1} \cdot e_{i2} \cdot \cdots \cdot e_{im} = \prod_{1 \leqslant k \leqslant m} e_{ik}. \tag{16}$$

The total error becomes zero when the distance between point $\mathbf{x}_i$ and any pattern is zero, i.e., $e_{ik} = 0$, and $E_i = 0$.

4. Parameter learning rules

Initially give random values to parameters of each pattern (line or hyperbola). For each input $\mathbf{x}_i$, we calculate the distance from $\mathbf{x}_i$ to each pattern. Then calculate the total error, E_i, of $\mathbf{x}_i$, and use gradient descent method to update parameters of each pattern to reach minimum error. Fig. 13 illustrates the change of parameters corresponding to the change of the pattern, i.e., pattern matches the data by learning. The parameters of line or hyperbola can be written as a parameter vector $\mathbf{p}_k$, and the learning $\mathbf{p}_k(t+1) = \mathbf{p}_k(t) + \Delta\mathbf{p}_k(t)$, where ($k = 1, 2, \ldots, m$), and by gradient descent method:

$$\Delta\mathbf{p}_k = -\beta \frac{\partial E_i}{\partial \mathbf{p}_k}, \tag{17}$$

where β is the learning rate. From Eq. (17) and by chain rules, $\Delta\mathbf{p}_k$ can be written as

$$\begin{aligned}\Delta\mathbf{p}_k &= -\beta\left(\frac{\partial E_i}{\partial e_{ik}}\right)\left(\frac{\partial e_{ik}}{\partial d_{ik}}\right)\left(\frac{\partial d_{ik}}{\partial \mathbf{p}_k}\right) \\ &= -\beta\left(\frac{E_i}{e_{ik}}\right)\left(\frac{2d_{ik}}{\sigma^2}\right)(1 - e_{ik})\left(\frac{\partial d_{ik}}{\partial \mathbf{p}_k}\right).\end{aligned} \tag{18}$$

We derive $\partial d_{ik}/\partial \mathbf{p}_k$ for hyperbola and line, respectively, as follows.

4.1. Learning rule for hyperbola

For the non-rotated kth hyperbola, or seismic reflection pattern, in Eq. (6), the parameter vector of hyperbola is $\mathbf{p}_k = [\,a_k \quad b_k \quad x_{0,k} \quad t_{0,k}\,]^{\mathrm{T}}$, and thus

$$\frac{\partial d_{ik}}{\partial \mathbf{p}_k} = \begin{bmatrix} \dfrac{\partial d_{ik}}{\partial a_k} & \dfrac{\partial d_{ik}}{\partial b_k} & \dfrac{\partial d_{ik}}{\partial x_k} & \dfrac{\partial d_{ik}}{\partial t_k} \end{bmatrix}. \tag{19}$$

From Eq. (10), the elements of $\partial d_{ik}/\partial \mathbf{p}_k$ are

$$\frac{\partial d_{ik}}{\partial a_k} = \frac{(-(b_k/a_k))((x_i - x_{0,k})/a_k)^2}{\sqrt{((x_i - x_{0,k}/a_k))^2 + 1}},$$

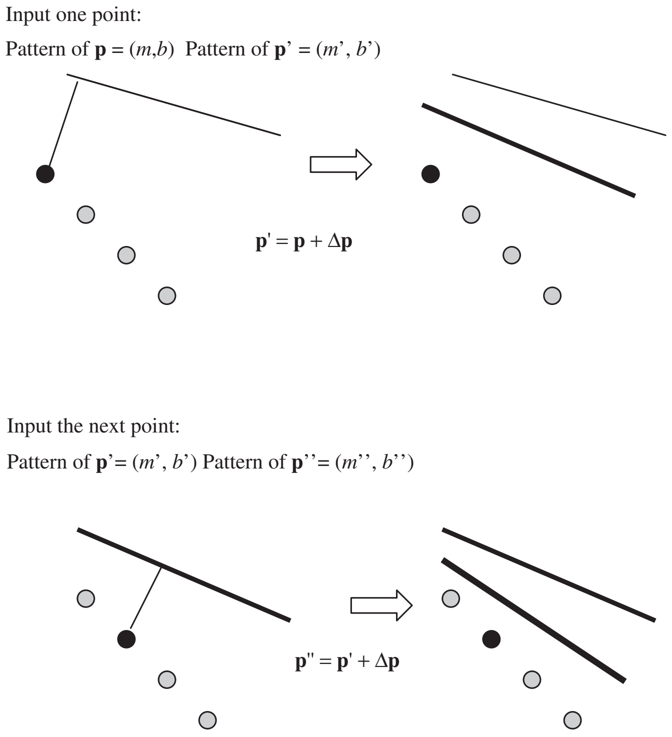

Input one point:
Pattern of **p** = (*m*,*b*) Pattern of **p**' = (*m*', *b*')
p' = **p** + Δ**p**
Input the next point:
Pattern of **p**'= (*m*', *b*') Pattern of **p**''= (*m*'', *b*'')
p'' = **p**' + Δ**p**

Fig. 13. Illustration of parameter learning.

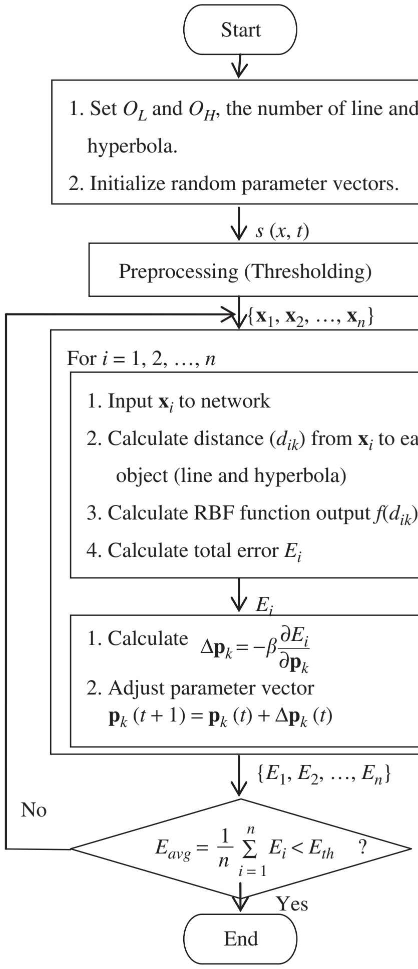

Start
1. Set O_L and O_H, the number of line and hyperbola.
2. Initialize random parameter vectors.
$s(x, t)$
Preprocessing (Thresholding)
$\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$
For $i = 1, 2, \ldots, n$
1. Input $\mathbf{x}_i$ to network
2. Calculate distance (d_{ik}) from $\mathbf{x}_i$ to each object (line and hyperbola)
3. Calculate RBF function output $f(d_{ik})$
4. Calculate total error E_i
E_i
1. Calculate $\Delta\mathbf{p}_k = -\beta\frac{\partial E_i}{\partial \mathbf{p}_k}$
2. Adjust parameter vector $\mathbf{p}_k(t+1) = \mathbf{p}_k(t) + \Delta\mathbf{p}_k(t)$
$\{E_1, E_2, \ldots, E_n\}$
No
$E_{avg} = \frac{1}{n}\sum_{i=1}^{n} E_i < E_{th}$?
Yes
End

Fig. 14. Flowchart of the learning system.

$$\frac{\partial d_{ik}}{\partial b_k} = \sqrt{\left(\frac{x_i - x_{0,k}}{a_k}\right)^2 + 1},$$
$$\frac{\partial d_{ik}}{\partial x_{0,k}} = \frac{(-(b_k/a_k))((x_i - x_{0,k})/a_k)}{\sqrt{((x_i - x_{0,k}/a_k))^2 + 1}},$$
$$\frac{\partial d_{ik}}{\partial t_{0,k}} = 1. \tag{20}$$

Then, from Eqs. (18) and (20), we have

$$\Delta p_k = [\,\Delta a_k \quad \Delta b_k \quad \Delta x_{0,k} \quad \Delta t_{0,k}\,]^{\mathrm{T}}$$
$$= -\beta\left(\frac{E_i}{e_{ik}}\right)\left(\frac{2d_{ik}}{\sigma^2}\right)(1 - e_{ik})\begin{bmatrix} \dfrac{(-(b_k/a_k))((x_i - x_{0,k})/a_k)^2}{\sqrt{((x_i - x_{0,k})/a_k)^2 + 1}} \\ \sqrt{\left(\frac{x_i - x_{0,k}}{a_k}\right)^2 + 1} \\ \dfrac{(-(b_k/a_k))((x_i - x_{0,k})/a_k)}{\sqrt{((x_i - x_{0,k})/a_k)^2 + 1}} \\ 1 \end{bmatrix}. \tag{21}$$

For seismic reflection wave from the horizontal reflection layer, we have $x_{0,k} = 0$ in Eq. (6). So the parameter vector is $\mathbf{p}_k = [\,a_k \quad b_k \quad t_{0,k}\,]^{\mathrm{T}}$, and by Eq. (21), which implies parameter adjustment:

$$\Delta \mathbf{p}_k = [\,\Delta a_k \quad \Delta b_k \quad \Delta t_{0,k}\,]^{\mathrm{T}}$$
$$= -\beta\left(\frac{E_i}{e_{ik}}\right)\left(\frac{2d_{ik}}{\sigma^2}\right)(1 - e_{ik})\begin{bmatrix} \dfrac{(-(b_k/a_k))((x_i - x_{0,k})/a_k)^2}{\sqrt{((x_i - x_{0,k})/a_k)^2 + 1}} \\ \sqrt{\left(\frac{x_i - x_{0,k}}{a_k}\right)^2 + 1} \\ 1 \end{bmatrix}. \tag{22}$$

4.2. Learning rule for line

For the kth line, the parameter vector is $\mathbf{p}_k = [\,m_k \quad b_k\,]^{\mathrm{T}}$. Thus,

$$\frac{\partial d_{ik}}{\partial p_k} = \left[\frac{\partial d_{ik}}{\partial m_k} \quad \frac{\partial d_{ik}}{\partial b_k}\right]^{\mathrm{T}}. \tag{23}$$

From Eq. (14), we can get:

$$\frac{\partial d_{ik}}{\partial m_k} = x_i \quad \text{and} \quad \frac{\partial d_{ik}}{\partial b_k} = 1. \tag{24}$$

Hence, from Eqs. (18) and (24), we have

$$\Delta \mathbf{p}_k = \left[\,\Delta m_k \quad \Delta b_k\,\right]^{\mathrm{T}}$$
$$= -\beta\left(\frac{E_i}{e_{ik}}\right)\left(\frac{2d_{ik}}{\sigma^2}\right)(1 - f(d_{ik}))\left[\,x_i \quad 1\,\right]^{\mathrm{T}}. \tag{25}$$

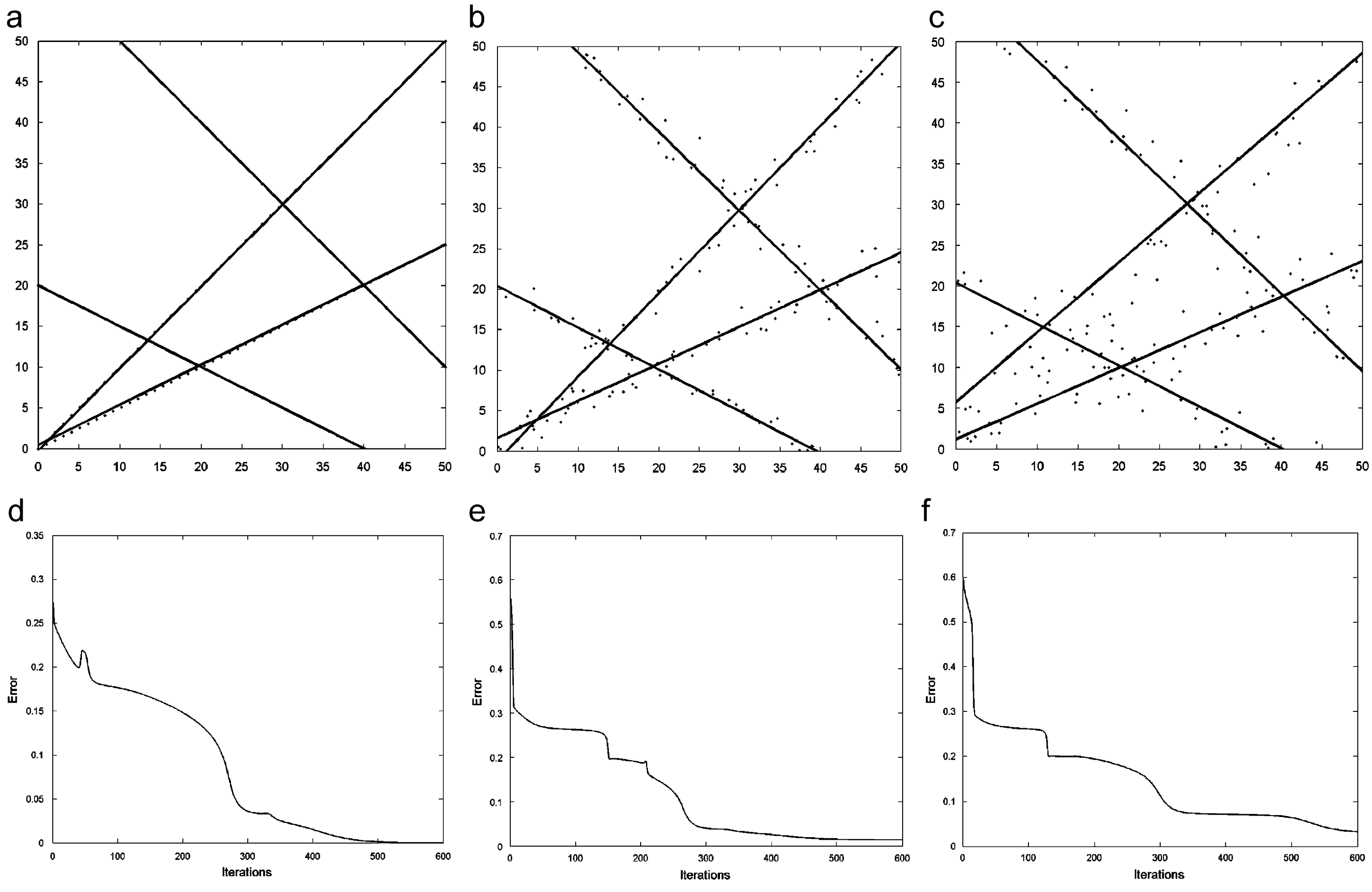


Fig. 15. Experiments on lines: (a) four lines, (b) data with Gaussian noise $N(0,1)$, (c) data with Gaussian noise $N(0,2)$, and (d), (e), (f) corresponding error plot with iterations.

4.3. Learning process

The flowchart of the learning system is shown in Fig. 14. Here, we use simple one-stage learning instead of two-stage learning [1], that is, all parameters are changed simultaneously. The number of patterns or objects, including lines and hyperbolas, O_L and O_H, must be given. Initially give random parameter vectors. Then input each data and adjust the parameter vector as in Eq. (21) or (22) and (25). One input data has one error. Then we calculate the average error for all input data. Finally, if the average error is less than a threshold, E_{th}, or the iterations reach the preset maximum iteration number, the learning stops.

5. Experiments

In the first part, several lines and hyperbolas are generated and detected. In the second part, we do the experiments on the simulated one-shot seismogram for the line detection of direct wave and the hyperbola detection of reflection wave. In the third part, we do the experiments on the detection of direct wave and the reflection wave in the real seismic data.

5.1. Experiments on the detection of line and hyperbolic patterns in simulated image data

We do the experiments on the detection of lines, hyperbolas, and both of them in Figs. 15–17. The learning rates of the center (x_0, y_0), the major and minor axes (a and b) of hyperbola, the slope (m) and intercept (b) of line are not the same and set by experiments. In experiments of part one, learning rates for the center is 1, for the major and minor axes of hyperbola is 0.5, for the slope of line is 0.01 and for the intercept of line is 1. The σ in Gaussian basis function is preset and is found to be related to the size of images. In experiment part one, σ is 10.

In Fig. 15(a), there are four lines, $O_L = 4$, with total 200 points. In Fig. 15(b), the data are disturbed by Gaussian random noise with zero mean and variance one, that is, $N(0,1)$. In Fig. 15(c), the Gaussian noise is $N(0,2)$. We see that lines in Fig. 15(a) and (b) are well detected even if Fig. 15(b) has noise data. But in Fig. 15(c) larger noise data, the detection result is affected by noise.

In Fig. 16(a), there are two hyperbolas (100 points), $O_H = 2$. And in Fig. 16(b) and (c), data are with Gaussian random noise $N(0,1)$, and $N(0,2)$, respectively. Results show that hyperbolas are detected correctly in Fig. 16(a)–(c).

In Fig. 17(a), there are two lines, $O_L = 2$, and two hyperbolas, $O_H = 2$, with total 200 points, and data with Gaussian random noise $N(0,1)$ and $N(0,2)$ in Fig. 17(b) and (c). Results show that hyperbolas are detected correctly in Fig. 17(a)–(c).

The parts (d), (e), and (f) of Figs. 15–17 are plots of error versus iterations for parts (a), (b), and (c) of Figs. 15–17, respectively.

5.2. Experiments on simulated seismic data

5.2.1. Horizontal reflection layer

The HTNN is applied to the simulated seismic data for line and hyperbolic detection. Fig. 8 is the simulated one-shot seismogram from the horizontal reflection layer where the depth of the

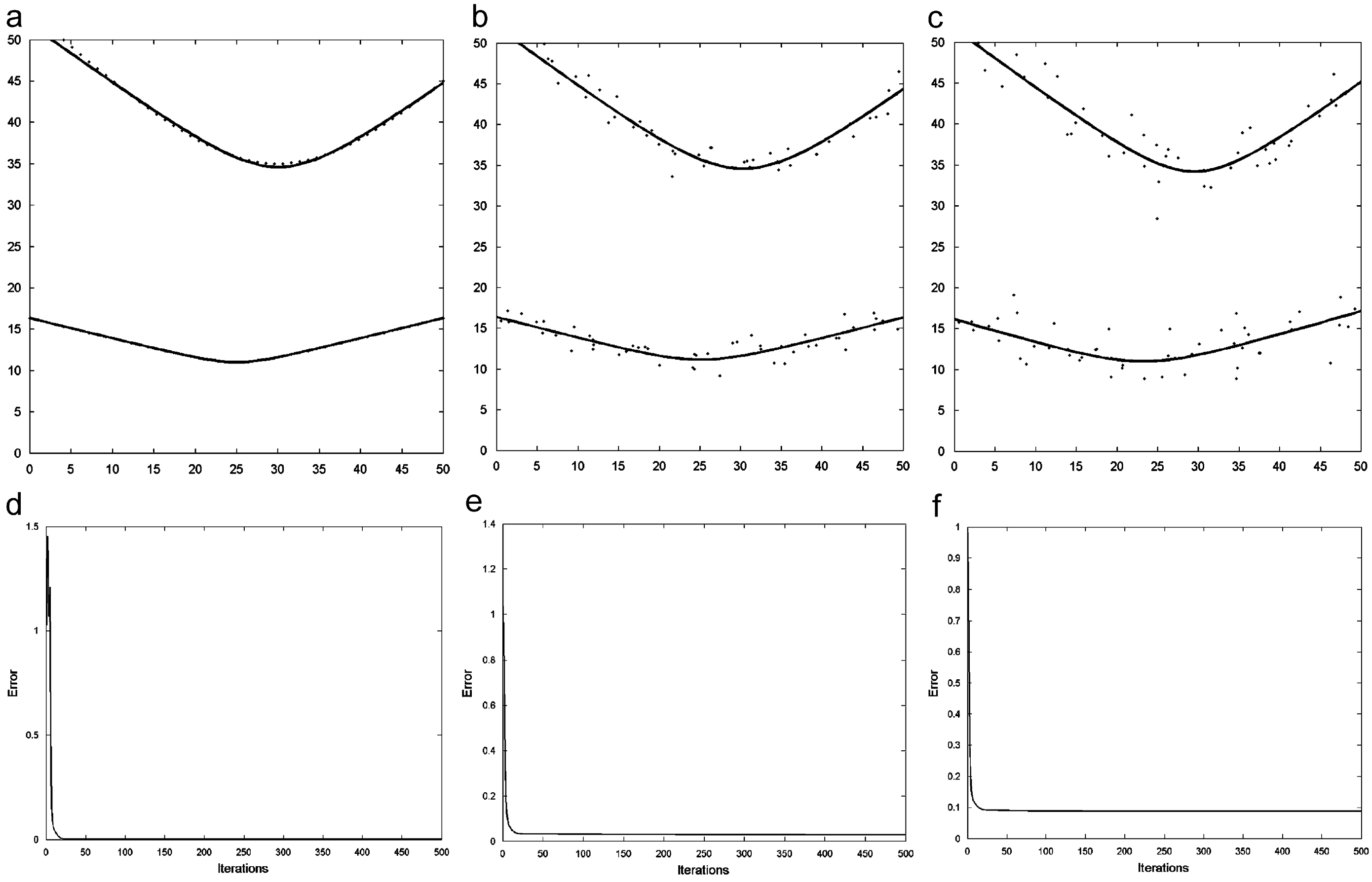


Fig. 16. Experiments on lines: (a) two hyperbolas, (b) data with Gaussian noise $N(0,1)$, (c) data with Gaussian noise $N(0,2)$, and (d), (e), (f) corresponding error plot with iterations.

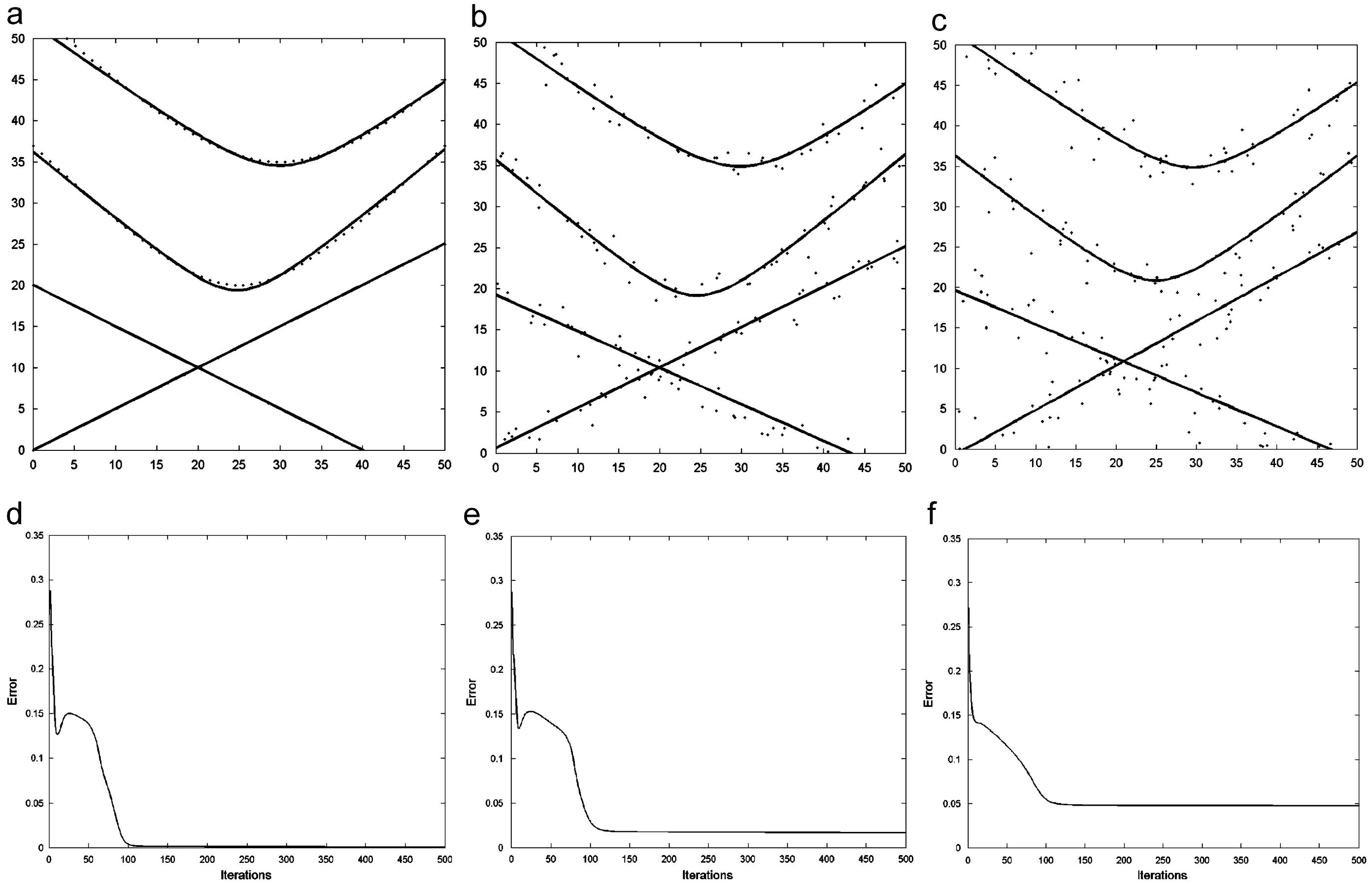


Fig. 17. Experiments on lines and hyperbolas: (a) two lines and two hyperbolas, (b) data with Gaussian noise $N(0,1)$, (c) data with Gaussian noise $N(0,2)$, and (d), (e), (f) corresponding error plot with iterations.

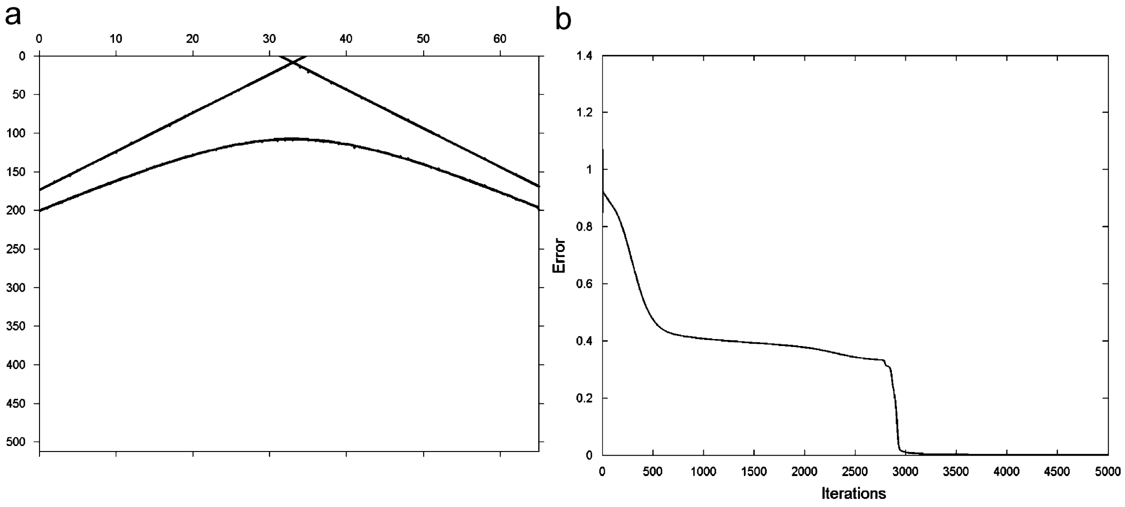


Fig. 18. (a) Line and hyperbola in Fig. 9 detected by HTNN. (b) Plot of error versus iterations.

reflection layer is 500 m and the velocity of the p-wave in the sedimentary rock is about 2500 m/s [11]. There are 65 receiving stations on both sides of explosion with 50 m between each two receiving stations. The sampling interval is 0.004 s. The impulse response is 25 Hz Ricker wavelet. Reflection coefficient is 0.2 and noise is band-passed noise, 10.2539–59.5703 Hz, with uniform distributed over $(-0.2, 0.2)$. The one-shot seismogram in Fig. 8 is first preprocessed by envelope processing, threshold processing, and peak detection processing. The preprocessing result is shown in Fig. 9. The image size is 512×65 where the origin is on the top-left corner with horizontal x-axis and vertical y-axis. These 129 points are then used as the input data to HTNN.

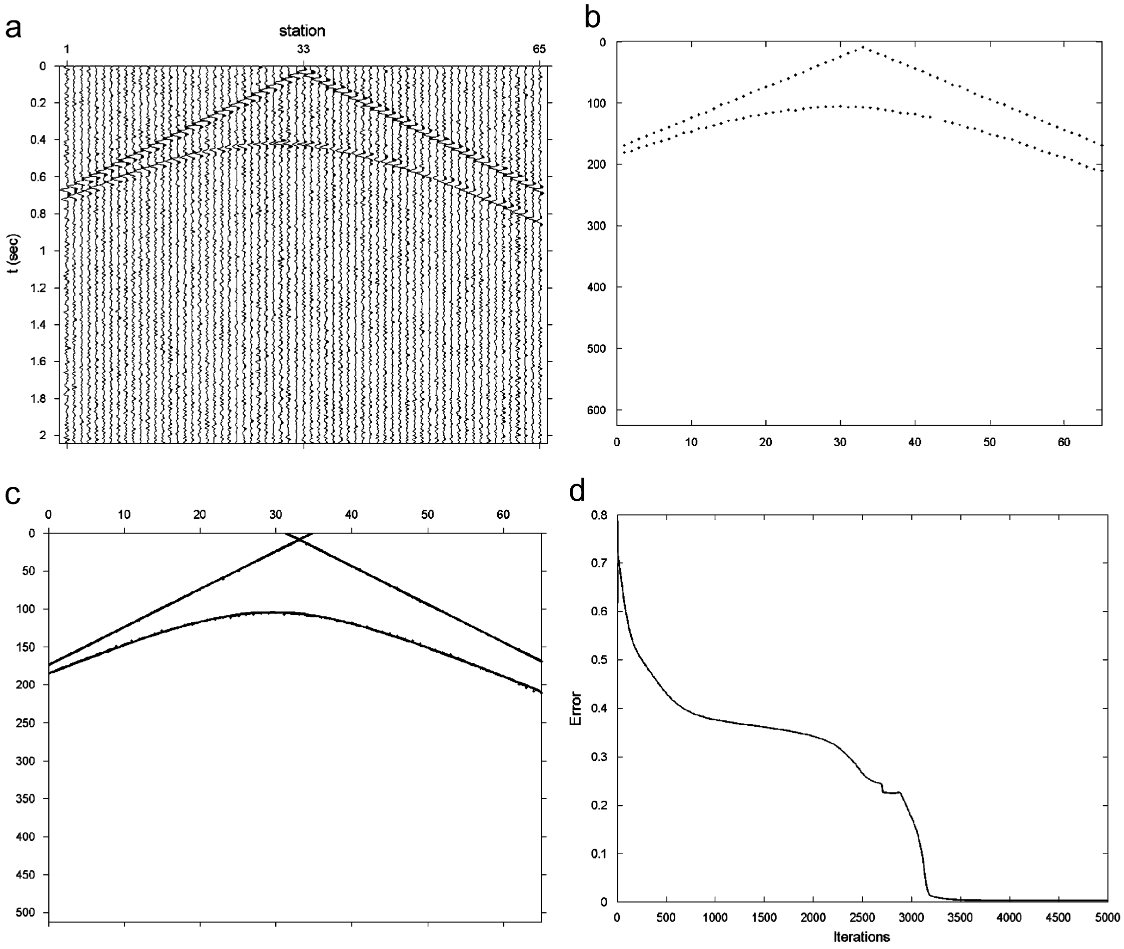


Fig. 19. (a) One-shot seismogram for dipping reflection layer. (b) Result of preprocessing. (c) Line and hyperbola detected by HTNN. (d) Plot of error versus iterations.

In this part of the experiment, learning rates for the center is 0.1, for the major and minor axes of hyperbola is 0.1, for the slope of line is 0.1 and for the intercept of line is 5. The σ in Gaussian basis function is 25. The preset numbers of lines and hyperbolas are $O_L = 2$ and $O_H = 1$. Fig. 18(a) shows the result of lines and the hyperbola detection in Fig. 9. And the plot of error versus iterations is shown in Fig. 18(b).

5.2.2. *Dipping reflection layer*

Fig. 19 shows the detection of direct and reflection wave from the dipping reflection layer by HTNN. Fig. 19(a) is the original one-shot seismogram. After preprocessing, the input data in Fig. 19(b) have 127 points. The detection results are in Fig. 19(c). And the error plot is shown in Fig. 19(d). The result is also good.

5.3. *Experiments on real seismic data*

The system is also applied to detect direct wave and reflection wave in real seismic data. We obtain data from Seismic Unix System developed by Colorado School of Mines [14].

The real data with the size 3100×48 shown in Fig. 20(a) is from Canadian Arctic, which has 48 traces and 3100 samples per trace with sampling interval 0.002 s. The horizontal axis is the trace number and the vertical axis is time t. After preprocessing [7], Fig. 20(b) shows the result of preprocessing with the threshold 0.15. We only choose 53 points with $y < 500$ which includes points from direct wave and first reflection wave as input data to the HTNN. Learning rates for the center is 0.1, for the major and minor axes of hyperbola is 10, for the slope of line is 1, and for the intercept of line is 100. The σ in Gaussian basis function is 80. The preset numbers of lines and hyperbolas are $O_L = 2$ and $O_H = 1$. Fig. 20(c) is the detection result of two lines and a hyperbola. The plot of the result on the original one-shot seismogram is shown in Fig. 20(d). The detected parameters of lines of direct wave and hyperbola of reflection wave in image space in Fig. 20(c) are listed in Table 1.

6. Conclusions

HTNN is adopted to detect the line pattern and the hyperbola pattern in image data, and is also adopted to detect the line pattern of direct wave and the hyperbola pattern of reflection wave in a one-shot seismogram. The objects can fit the data. The parameter learning rules are derived by gradient descent method to minimize the total error. We define the vertical time difference as the distance from point to hyperbola that makes the learning feasible. In the experiments on the pattern detection of the line and hyperbola, patterns are well-detected even if there are noises. The method can tolerate certain level of noise. For the data with large noise, the method may suffer from the local optimal problem. In the experiments on seismic data, the detection results in the line pattern of direct wave and the hyperbola pattern of

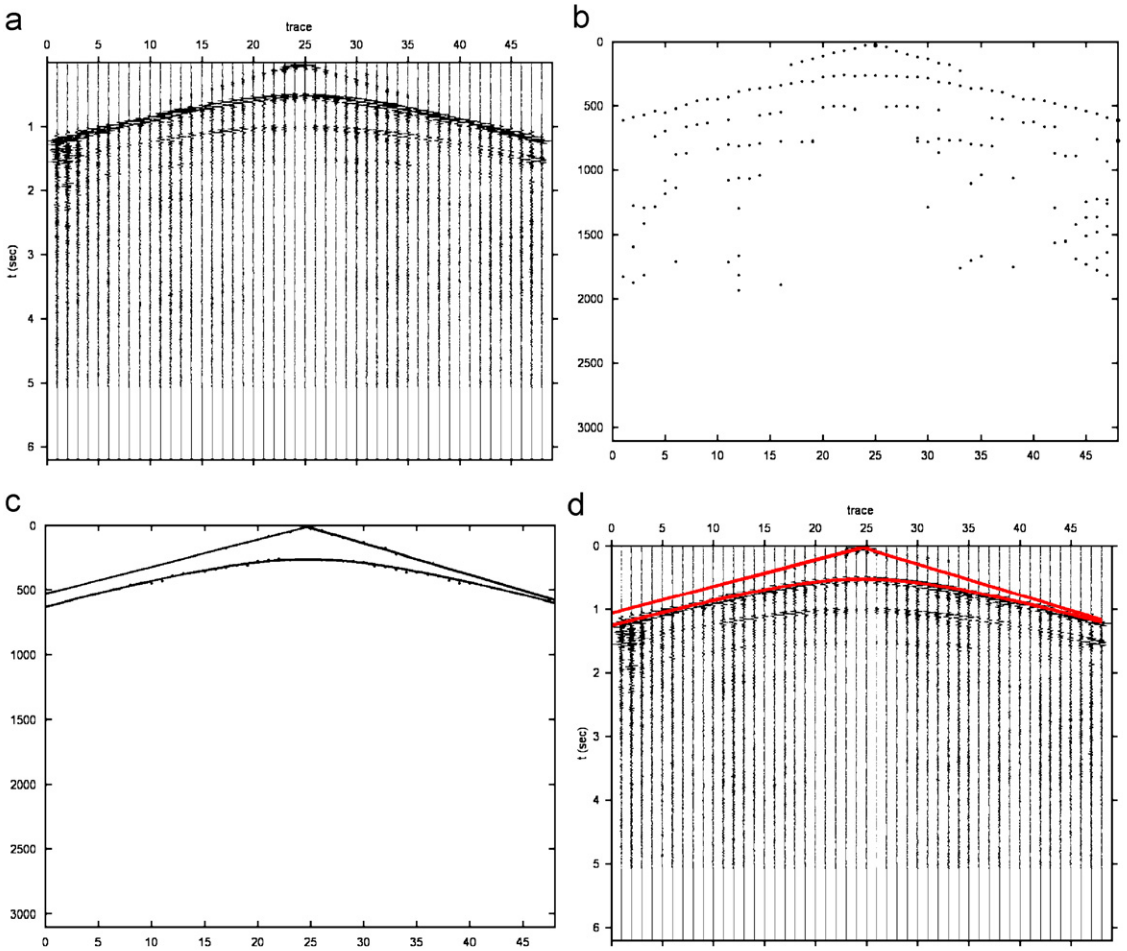


Fig. 20. (a) One-shot seismogram from Canadian Artic. (b) Result of envelope threshold and peak detection process. (c) Detection result of peaks with $y<500$. (d) Detected lines and the hyperbola superimposed on the one-shot seismogram.

Table 1
Detected parameters in Fig. 20(c) in image space 3100 × 48

Hyperbola of reflection wave	x_0	y_0	a	b
	24.69	23.19	10.65	240.96
Lines of direct wave	m			b
	24.28 −21.08			−583.48 534.24

reflection wave are good. The detection results can improve seismic interpretation and further seismic data processing.

In the experiments, the detection results depend on the learning rate and parameter σ in the Gaussian basis function. These parameters are set heuristically. Further research may be done to justify why these selected values provide good results in the experiments.

Acknowledgments

The authors would like to thank the Colorado School of Mines for the use of Seismic Unix System and the real data. This work was supported in part by the National Science Council, Taiwan, under NSC-94-2213-E-009-133 and NSC-95-2221-E-009-221.

References

[1] J. Basak, A. Das, Hough transform networks: learning conoidal structures in a connectionist framework, IEEE Trans. Neural Netw. 13 (2) (2002) 381–392.
[2] J. Basak, A. Das, Hough transform network: a class of networks for identifying parametric structures, Neurocomputing 51 (2003) 125–145.
[3] G.L. Dempsey, E.S. McVey, A Hough transform system based on neural networks, IEEE Trans. Ind. Electron. 39 (1992) 522–528.
[4] M.B. Dobrin, Introduction to Geophysical Prospecting, third ed., McGraw-Hill, 1976.
[5] R.O. Duda, P.E. Hart, Use of the Hough transform to detect lines and curves in pictures, Commun. Assoc. Comput. Mach. 15 (1972) 11–15.
[6] P.V.C. Hough, Method and Means for Recognizing Complex Patterns, US Patent 3069654, 1962.
[7] K.Y. Huang, K.S. Fu, T.H. Sheen, S.W. Cheng, Image processing of seismograms: (A) Hough transformation for the detection of seismic patterns. (B) Thinning processing in the seismogram, Pattern Recognit. 18 (6) (1985) 429–440.
[8] K.Y. Huang, T.H. Sheen, S.W. Cheng, Z.S. Lin, K.S. Fu, Seismic image processing: (I) Hough transformation, (II) thinning processing, (III) linking processing, handbook of geophysical exploration: section I, Seism. Explor., Pattern Recognit. Image Process. 20 (1987) 79–109.
[9] K.Y. Huang, J.D. You, K.J. Chen, H.L. Lai, A.J. Don, Hough transform neural network for seismic pattern detection, in: International Joint Conference on Neural Networks, July 16–21, Vancouver, Canada, 2006, pp. 4670–4675.
[10] J. Illingworth, J. Kittler, Survey: a survey of the Hough transform, Comput. Vis., Graph., Image Process. 44 (1988) 87–116.

[11] P. Kearey, M. Brooks, An Introduction to Geophysical Exploration, Blackwell Science Publications, Oxford, 1991.
[12] V.F. Leavers, Survey: which Hough transform, Comput. Vis., Graph., Image Process. 58 (2) (1993) 250–264.
[13] M.M. Slotnick, Seismic Computing, The Society of Exploration Geophysicists, 1959.
[14] O. Yilmaz, Seismic Data Processing, The Society of Exploration Geophysicists, Tulsa, 1987.

Kou-Yuan Huang received the B.S. in physics (1973) and M.S. in geophysics (1977) from the National Central University, Taiwan, and M.S.E.E. (1980) and Ph.D. in electrical and computer engineering (1983) from Purdue University. From September 1983 to August 1988, he joined the faculty in the Department of Computer Science, University of Houston-University Park. Now he is the Professor at the Department of Computer Science at National Chiao Tung University, Hsinchu, Taiwan. He widely published papers in journals: Geophysics, Geoexploration, Pattern Recognition, IEEE Transactions on Geoscience and Remote Sensing, Computer Processing of Oriental Languages, etc. He has authored two books, Neural Networks and Pattern Recognition, Weikeg Publishing Co., Taipei, Taiwan, 2000, 2nd ed., 2003, and Syntactic Pattern Recognition for Seismic Oil Exploration, World Scientific Publishing Co., Singapore, in Series in Machine Perceptron and Artificial Intelligence, Vol. 46, 2002. His major contributions are in the areas of seismic pattern recognition using signal and image processing, statistical, syntactic, neural networks, fuzzy logic, and genetic methods.

Kai-Ju Chen received the B.S. in computer science (2005) and M.S. in multimedia engineering (2007) from the National Chiao Tung University, Taiwan. His research interests are neural network, pattern recognition, and signal processing.

Jiun-Der You received the M.S. in computer science (2004) from the National Chiao Tung University, Taiwan.

An-Ching Tung received the B.S. in computer science (2005) from National Central University, Taiwan, and M.S. in multimedia engineering (2007) from the National Chiao Tung University, Taiwan.