Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007

Simulated Annealing for Pattern Detection and Seismic Application

Kai-Ju Chen and Kou-Yuan Huang

Abstract—**Simulated annealing algorithm is adopted to detect the parameters of lines, circles, ellipses, and hyperbolic patterns. We define the distance from a point to a pattern such that the detection becomes feasible, especially in hyperbola. The proposed simulated annealing parameter detection system has the capability to find a set of parameter vectors with global minimal error to the input data. Using average minimum distance, we propose a method to determine the number of patterns automatically. Experiments on the detection of lines, circles, ellipses, and hyperbolas in images are quite successful. The detection system is also applied to detect the line pattern of direct wave and the hyperbolic pattern of reflection wave in the simulated one-shot seismogram. The results are good and can improve seismic interpretations and further seismic data processing.**

I. Introduction

Traditionally, Hough transform (HT) was used to detect the parametric patterns such as lines and circles by mapping the data in the image into the parameter space and detecting the peak (maximum) in the parameter space [1]-[2]. The coordinates of the peak in parameter space corresponds to a pattern in the image space.

Seismic pattern recognition plays an important role in oil exploration. In 1985, Huang et al had applied the HT to detect line pattern of direct wave and hyperbolic pattern of reflection wave [3]-[4] in a one-shot seismogram [5]. However, it was not easy to determine the peak in the parameter space and the large memory requirement was also a serious problem.

The Hough transform neural network (HTNN) was developed to solve the HT problem [6]. It was designed to detect lines, circles, and ellipses by gradient descent method to minimize the distance between patterns and points. HTNN was also adopted to detect line and hyperbola in a one-shot seismogram by Huang et al [7]. The iterative method required less memory, but it suffered from local minimum problem.

Simulated annealing (SA) was first proposed by Kirkpatrick in 1983 [8]. The algorithm simulates the procedure of the substance frozen from melt to form a perfect crystal or low-energy state by careful annealing. The Metropolis criterion which conditionally allows the state of the system to higher energy condition is the key of the SA algorithm to reach the global minimum.

Here, we take the advantage of global optimization in SA to minimize the distance for detecting parametric patterns: lines, circles, ellipses, and hyperbolas. Also the proposed detection system is applied to the detection of line pattern of direct wave and hyperbolic pattern of reflection wave in the one-shot seismogram.

II. Detection System

Fig. 1 is a system overview. The detection system takes the N data as the input, followed by the SA parameter detection system to detect a set of parameter vectors of K patterns. After convergence, patterns are recovered from the detected parameter vectors.

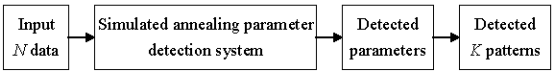

Fig. 1. System overview.

SA parameter detection system consists of two main parts: 1. definition of system error (energy, distance), and 2. SA algorithm for determination of the parameter vectors with minimum error. To obtain the system error, we calculate the error or the distance from a point to patterns, and then combine the errors from all points to patterns and to get the system error.

A. Parametric Patterns

1) Equation of the Parametric Pattern

Circle, ellipse, and hyperbola can be expressed by its center $\mathbf{c} = [c_x, c_y]^T$, pattern matrix

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad (1)$$

and the size r [6]. In the vector form, a parameter vector, $\mathbf{p} = [c_x, c_y, a_{11}, a_{12}, a_{21}, a_{22}, r]^T$, represents a pattern, the equation of pattern is

$$Q(\mathbf{x}) = (\mathbf{x}-\mathbf{c})^T \mathbf{A}(\mathbf{x}-\mathbf{c}) - r^2 = 0 \quad (2)$$

where image data $\mathbf{x}=[x, y]^T$.

An ellipse has a symmetric positive definite matrix $\mathbf{A}$ [6] which has the property of $\det(\mathbf{A})>0$ and $a_{11}>0$; circle has $\det(\mathbf{A})=1$. But a North-South opening hyperbola has a symmetric negative definite matrix $\mathbf{A}$ which has $\det(\mathbf{A})<0$. If $\det(\mathbf{A})=0$, (2) represents two parallel lines. A line can be approximate to a segment of an ellipse or the asymptote of a hyperbola. The properties of $\det(\mathbf{A})$ will affect the initial setting of the elements of $\mathbf{A}$ in the SA detection system.

This work was supported in part by the National Science Council, Taiwan, under NSC 95-2221-E-009-221.

Kou-Yuan Huang is with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan. (Corresponding author e-mail: kyhuang2@cs.nctu.edu.tw)

Kai-Ju Chen is with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan. (e-mail: chenkaiju@gmail.com)

2) Distance from a Point to a Pattern

Here the detected patterns include line, circle, ellipse, and hyperbola, the distance from a point $\mathbf{x}_i$ to the kth pattern is defined as

$$d_k(\mathbf{x}_i) = \left| \sqrt{\left| (\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{A}_k (\mathbf{x}_i - \mathbf{c}_k) \right|} - r_k \right| \quad (3)$$

with $|\det(\mathbf{A}_k)|=1$. For hyperbola, $\mathbf{A}_k$ is negative definite, $(\mathrm{x}_i\text{-}\mathrm{c}_k)^T\mathrm{A}_k(\mathrm{x}_i\text{-}\mathrm{c}_k)$ may be negative. So we take absolute value before square root. This definition makes computation feasible, especially in hyperbolic detection.

B. Distance from a Point to K Patterns

Error or distance from a point to the patterns is defined as the product of the distances from the point to all patterns. The error of the ith point $\mathbf{x}_i$ is

$$E_i = E(\mathbf{x}_i) = d_1(\mathbf{x}_i) d_2(\mathbf{x}_i) \ldots d_k(\mathbf{x}_i) \ldots d_K(\mathbf{x}_i), \quad (4)$$

where K is the total number of patterns. If the point is on any pattern, the error of this point will be zero. Fig. 2 shows the error of a point to all patterns. The distance layer computes the distance from a point to each pattern by (3), and the error layer outputs the error from a point to all patterns by (4).

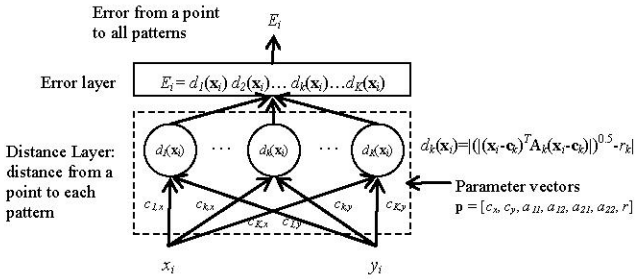

Fig. 2. Distance from a point to all patterns. i is the index of the input point. k is the index of the pattern, and K is the number of patterns.

C. Error from N input points to K Patterns in the System

Fig. 3 illustrates the error or energy of the system from N input points to K patterns. The error or energy of the system is defined as the average of the error of points powered by reciprocal of the number of patterns,

$$E = \left(\frac{1}{N} \sum_{i=1}^{N} E_i \right)^{1/K} \quad (5)$$

The use of power $1/K$ is to keep the energy E in some magnitude instead of growing as K increasing.

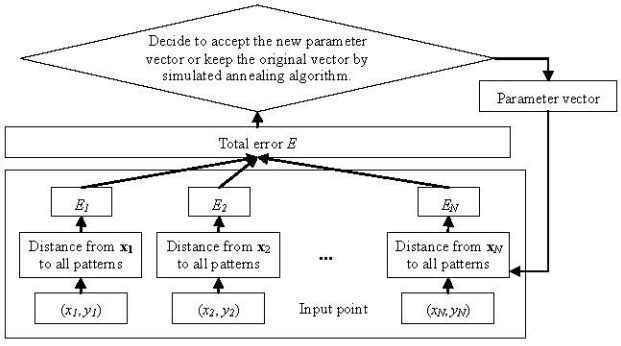

Fig. 3. Total error of the system and simulated annealing procedure.

D. Simulated Annealing Parameter Detection System

We use SA to detect the parameter vector of each pattern. Our goal is to find a set of parameter vectors that can globally minimize the error of the system. Using the temperature decreasing function $T(t)$

$$T(t) = T_{\max} / \ln(1+t) \quad \text{for } t=1, 2, 3, \ldots, \quad (6)$$

can converge to global minimum with probability 1 [9].

Adjusting all parameters at one time is not efficient in convergence [6]. We use three steps in adjusting parameters also. The adjusting order is the center, the shape matrix **A,** and then the size r. In order to get fast convergence, we use the accepted counts to change the adjusting amount [10]. The algorithm is in the following.

Algorithm: SA to detect parameter vectors of K patterns.

Input: N points in an image.

Output: A set of detected K parameter vectors.

Steps:

Step 1: Initialization.

At initial step $t=1$, choose $T(1)=T_{\max}/\ln(2)$ at high temperature, and define the temperature decreasing function

$T(t) = T_{\max} / \ln(1+t)$ for $t=1, 2, 3, \ldots,$

Initialize parameter vectors $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_k, \ldots, \mathbf{p}_K$, where $\mathbf{p}_k = [c_{k,x}, c_{k,y}, a_{k,11}, a_{k,12}, a_{k,21}, a_{k,22}, r_k]^T$, one $\mathbf{p}$ is for one pattern, and set $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_k, \ldots, \mathbf{p}_K)$.

Calculate energy $E(\mathbf{P})$ as (5).

Step 2: Randomly change parameter vectors and decide the new parameter vectors in the same temperature.

For m = 1 to Nr (adjust the step size Nr times in a temperature)

For j = 1 to Ns (adjust the step size per Ns trials)

For k = 1 to K (k is the index of the pattern)

(a) Randomly change the center of the kth pattern:

$$[c'_{k,x} \;\; c'_{k,y}]^T = [c_{k,x} \; c_{k,y}]^T + s_{k,c} \mathbf{v}_{k,c} \quad (7)$$

where $\mathbf{v}_{k,c}$ is a 2×1 random vector with its elements are uniformly distributed over (0, 1), and $s_{k,c}$ is the step size. Now, $\mathbf{p}'_k = [c'_{k,x}, c'_{k,y}, a_{k,11}, a_{k,12}, a_{k,21}, a_{k,22}, r_k]^T$, and $\mathbf{P}' = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}'_k, \ldots, \mathbf{p}_K)$.

Calculate the new energy $E(\mathbf{P}')$ from N points to K patterns. Using Metropolis criterion decides whether or not to accept $\mathbf{P}'$. Metropolis criterion is a rule [11]: for the new energy less than or equal to the original one, $\Delta E = E(\mathbf{P}') - E(\mathbf{P}) \le 0$, accept $\mathbf{P}'$. Otherwise, new energy is higher than the original one, $\Delta E = E(\mathbf{P}') - E(\mathbf{P}) > 0$. In this case, compute $prob = \exp(-\Delta E/T(t))$, and generate a random number x uniformly distributed over (0, 1). If $prob \ge x$, accept $\mathbf{P}'$; otherwise, reject it, and keep $\mathbf{P}$.

If $\mathbf{P}'$ is accepted, increase the number of acceptance for the adjusting center of the kth pattern, $N_{k,c} = N_{k,c}+1$.

(b) Randomly change the shape parameters:

$$[a'_{k,11} \; a'_{k,12} \; a'_{k,21} \; a'_{k,22}]^T = [a_{k,11} \; a_{k,12} \; a_{k,21} \; a_{k,22}]^T + s_{k,A} \mathbf{v}_{k,A} \quad (8)$$

where $\mathbf{v}_{k,A}$ is a 4×1 random vector with its elements are uniformly distributed over (0, 1) and $s_{k,A}$ is the step size. Then, $[a'_{k,11}, a'_{k,12}, a'_{k,21}, a'_{k,22}]^T$ is normalized by the

square root of |det(**A**')|. Now, $\mathbf{P}'_k = [c_{k,x}, c_{k,y}, a'_{k,11}, a'_{k,12}, a'_{k,21}, a'_{k,22}, r_k]^T$, and $\mathbf{P}'=(\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}'_k, \ldots, \mathbf{p}_K)$.

Similar to Step 2(a), calculate the new energy $E(\mathbf{P}')$ from N points to K patterns. Using Metropolis criterion decides whether or not to accept **P**'. If **P**' is accepted, increase the number of acceptance for the adjusting shape of the kth pattern, $N_{k,A} = N_{k,A}+1$.

(c) Randomly change the size:

$$r'_k = r_k + s_{k,r} v_{k,r} \tag{9}$$

where $v_{k,r}$ is a random number uniformly distributed over (0, 1) and $s_{k,r}$ is the step size. Now, $\mathbf{P}'_k = [c_{k,x}, c_{k,y}, a_{k,11}, a_{k,12}, a_{k,21}, a_{k,22}, r_k']^T$, and $\mathbf{P}'=(\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}'_k, \ldots, \mathbf{p}_K)$.

Similar to Step 2(a), calculate the new energy $E(\mathbf{P}')$ from N points to K patterns. Using Metropolis criterion decides whether or not to accept **P**'. If **P**' is accepted, increase the number of acceptance for the adjusting size of the kth pattern, $N_{k,r} = N_{k,r}+1$.

End for k

End for j

(d) Adjust the step size $s_{k,c}$, $s_{k,A}$, and $s_{k,r}$ in (7), (8), and (9). Compute the accepted rate for $s_{k,c}$, $s_{k,A}$, and $s_{k,r}$, k = 1, 2, …K:

$$\begin{aligned} rate_{k,c} &= N_{k,c} / Ns \\ rate_{k,A} &= N_{k,A} / Ns \\ rate_{k,r} &= N_{k,r} / Ns \end{aligned} \tag{10}$$

Calculate function value

$$f(rate) = \begin{cases} (1 + 2(rate - 0.6)/0.4) & \text{if } rate > 0.6 \\ \dfrac{1}{(1 + 2(0.4 - rate)/0.4)} & \text{if } rate < 0.4 \\ 1 & \text{if } 0.4 \le rate \le 0.6 \end{cases} \tag{11}$$

And get the adjusted step size

$$s = s \cdot f(rate) \tag{12}$$

where $rate$ is for $rate_{k,c}$, $rate_{k,A}$, and $rate_{k,r}$, and s is for $s_{k,c}$, $s_{k,A}$, and $s_{k,r}$.

End for m

Step 3: Cool the System.

Decrease temperature T according to the cooling function (6), and repeat Step 2, and 3 until the temperature is low enough, for examples, repeat 1,000 times.

In Step 2(d), adjust the step size $s_{k,c}$, $s_{k,A}$, and $s_{k,r}$ in (7), (8), and (9). Adjusting **P** to **P**' must get fast convergence. As in [10], the step sizes $s_{k,c}$, $s_{k,A}$, and $s_{k,r}$, k = 1, 2, …, K, are adjusted to keep a 1:1 ratio between accepted and rejected parameters. The large accepted rate in (10) means that the randomly changed **P**' is too close to the old **P**, so expand the step size. Contrarily, the small accepted rate implies the new parameter **P**' is too far from the old **P**, so the step size must be shrunk to avoid too many rejections. As in [10], the function to adjust the step size is in (11) and shown in Fig. 4. The step size is adjusted as $s = s \cdot f(rate)$. This adjustment in step size $s_{k,c}$, $s_{k,A}$, and $s_{k,r}$ can control the accepted rate approximately between 0.4 and 0.6.

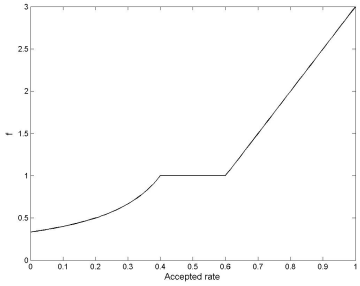

Fig. 4. Function for step size adjustment.

III. Implementation and Experimental Results

We do the experiments on simulated pattern detections in images with size 50×50. Because the properties of the pattern matrix A are different, to detect lines, circles, and ellipses, we use (14) and (15) in the following in order to meet the positive semi-definite matrix A. But, in hyperbolic pattern detection, we use (2) and (3). We also do the experiments on detecting line pattern of direct wave and hyperbolic pattern of reflection wave in a simulated one-shot seismogram. The effect of step size adjustments is discussed. And the final is the method for estimating the number of patterns.

A. Detection of lines, Circles, and Ellipses

For line, circle, and ellipse, matrix **A** is symmetric positive semi-definite and det(**A**) ≥ 0. A matrix **B** is used with |det(**B**)|=1,

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \tag{13}$$

and $\mathbf{A}=\mathbf{B}\mathbf{B}^T$ [6]. Thus, we can rewrite (2) to

$$Q(\mathbf{x}) = (\mathbf{x}-\mathbf{c})^T (\mathbf{B}\mathbf{B}^T)(\mathbf{x}-\mathbf{c}) - r^2 = 0 \tag{14}$$

and the distance measure (3) becomes

$$d(\mathbf{x}) = \left| \sqrt{|(\mathbf{x}-\mathbf{c})^T (\mathbf{B}\mathbf{B}^T)(\mathbf{x}-\mathbf{c})|} - r \right|. \tag{15}$$

Here, b_{12} is assumed to be equal to b_{21} to reduce the size of parameter vector. The parameter vector is

$$\mathbf{p} = [c_x, c_y, b_{11}, b_{22}, b_{12}, r]^T \tag{16}$$

size $r \ge 0$,

and (8) becomes

$$[b'_{k,11}\ b'_{k,12}\ b'_{k,12}\ b'_{k,22}]^T = [b_{k,11}\ b_{k,12}\ b_{k,12}\ b_{k,22}]^T + s_{k,A}\mathbf{v}_{k,A}, \tag{17}$$

and then divided by the square root of |det(**B**')|.

In initial stage, c_x and c_y are randomly distributed over (0, 50), r_k=1, b_{11}=1, b_{22}=1, and b_{12}=0. The cooling function is as (6) with a high enough temperature, $T_{\max}$=0.2. We adjust step size per Ns=20 as suggested in [10], and repeat N_r=5 times, so there are totally 100 trials in the same temperature. The temperature decreases 1,000 times to T=0.0289, and this temperature is low enough.

We use (14) and (15) to detect line. So a line can be considered as a segment of a long ellipse. Fig. 5 shows the results of detecting lines. There are two lines in each figure, and each line has 50 points. Data are disturbed by Gaussian noise N(0, 0.5)×N(0, 0.5). The error vs. cooling cycles shows that the error quickly decreases in the first few cycles and

goes toward lower energy gradually.

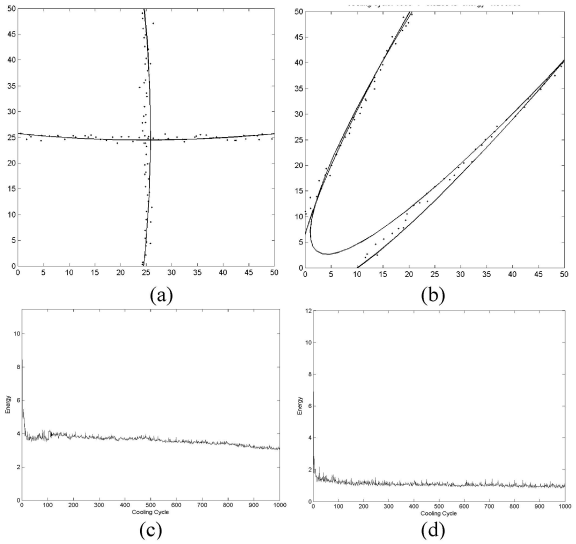

Fig. 5. Detection of lines − (a): 2 cross lines with noise. (b): 2 lines with noise. (c)-(d): corresponding error vs. cooling cycle of (a) and (b).

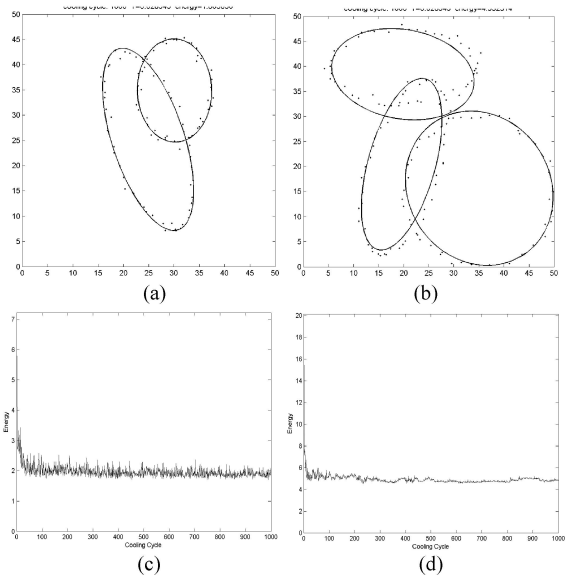

Fig. 6. Detection of circles and ellipses − (a): 2 ellipses with noise. (b): 2 ellipses and a circle with noise. (c)-(d): corresponding plot of error vs. cooling cycle of (a) and (b).

Result of detecting circles and ellipses are shown in Fig. 6. Each pattern has 50 points with Gaussian noise N(0, 0.5)×N(0, 0.5). Figures of energy vs. cooling cycles are also shown.

B. Detection of Hyperbolas

For a hyperbola, matrix **A** in (2) is negative definite and det(A)<0. Moreover, in a seismogram, hyperbola is always North-South vertical opening and non-rotated patterns. So we constrain the parameters to be a_{11}<0, a_{12}=0, a_{21}=0, and a_{22}>0, and size r≥0. Then, the parameter vector becomes

$$\mathbf{p} = [c_x, c_y, a_{11}, a_{22}, r]^T \quad (18)$$

and the distance calculation is in (3).

We initialize that c_x and c_y are randomly distributed over (0, 50), and a_{11}= -1, a_{22}=1, r=0 for hyperbolic pattern detection, and the cooling schedule is the same as the previous setting.

Fig. 7 shows the results of hyperbolic pattern detection, where Fig.7 (a) has 187 points and Fig. 7 (b) has 146 points. Each data is with Gaussian noise N(0, 0.5)×N(0, 0.5).

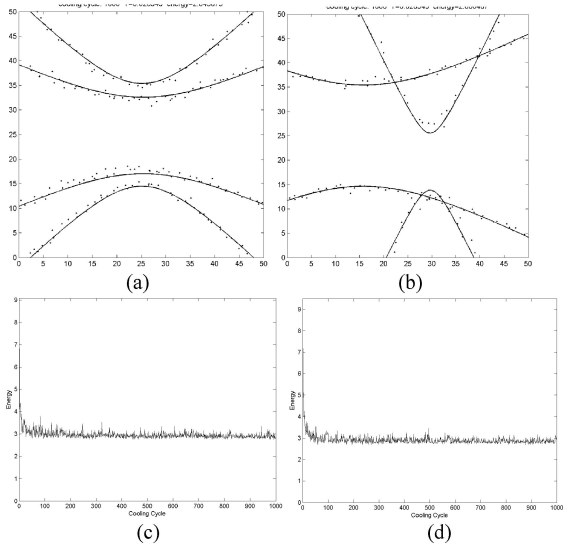

Fig. 7. Detection of hyperbolas − (a)-(b): 2 hyperbolas with noise. (c)-(d): corresponding plot of error vs. cooling cycle of (a) and (b).

C. Seismic Applications

Experiments have two cases: horizontal reflection layer and dipping reflection layer. Two lines are the asymptote of the hyperbola [3], and the asymptote is a hyperbola with the same shape but small size or size zero. So the asymptote can be treated as a hyperbola.

Fig. 8 (a) shows a one-shot seismogram from horizontal reflection layer. The two-sided one-shot seismogram has 65 receiving stations with 50 meters between each others, and 512 samples with sampling interval 0.004 seconds. The one-shot seismogram is first preprocessed by envelope processing and thresholding [5] as in Fig. 8 (b). The image size is 512×65. The points are then used as the input to the parameter detection system. The initial parameter **c** = (20, 20), a_{11}=−1, a_{22}=1, and r=0. The cooling function is as (6) with a high enough temperature, $T_{\max}$=0.4. The cooling schedule is Ns=20 for the step adjustment, and repeat Nr=5 times in a temperature. The temperature decreases 1,000 times. The result and the error plot are shown in Fig. 8 (c) and (d). The detection of direct wave and reflection wave from the dipping reflection layer is shown in Fig. 9.

Table I and II list the detected parameters for the above two seismic experiments. Comparing parameters of direct wave and hyperbolic wave in each experiment, the matrices **A** of both patterns are almost the same. Direct wave pattern is the asymptote of a hyperbola.

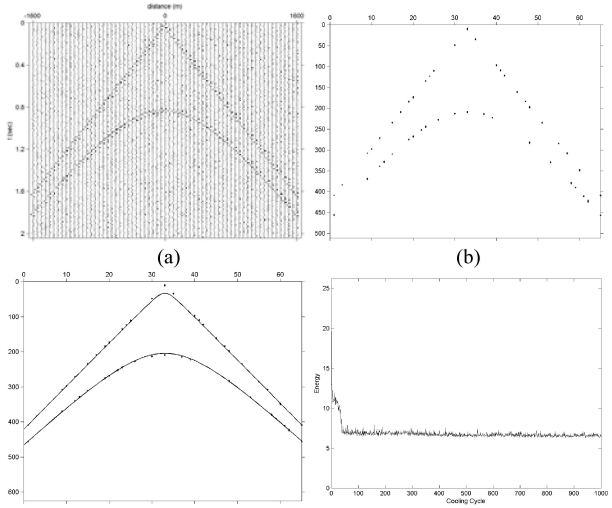

Fig. 8. Detection of seismic patterns − (a): simulated one-shot seismogram (horizontal reflection layer). (b) after envelope processing and thresholding. (c): detection result. (d): plot of error vs. cooling cycle.

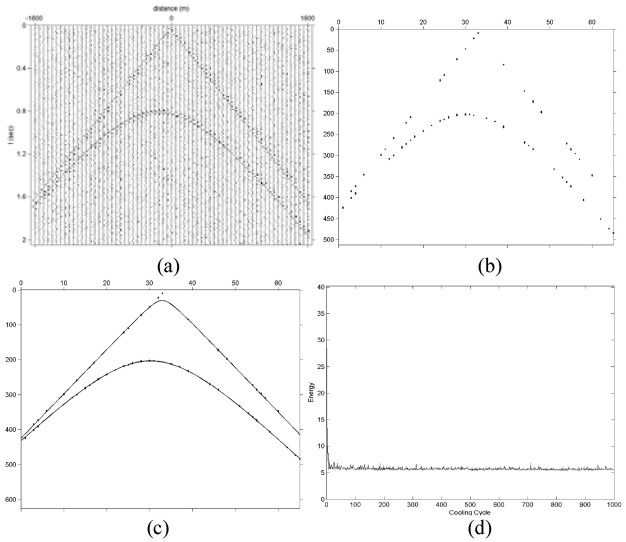

Fig. 9. Detection of seismic patterns − (a): simulated one-shot seismogram (dipping reflection layer). (b): after envelope processing and thresholding. (c): detection result. (d): plot of error vs. cooling cycle.

D. Step Size Adjustments

In the algorithm, we adjust the step size per *Ns* trials. The purpose is to get the fast convergence by choosing the appropriate step size.

A simple example compares the results of fixed step size in Fig. 10 and variable step size in Fig. 11. Here, the cooling function is the same as in the previous Section III *A*. But, the initial center is (0, 0) instead of random number, and the initial step size for all parameters are 0.1.

After 50 cooling cycles, the result of fixed step size in Fig. 10(a) is far from data, but the result of variable step size nearly matches data. The error of the system at the 50*th* cooling cycle is 2.4179 in Fig. 10(b) and 0.3613 in Fig. 11(b). Also note that, Fig. 10(c), the accepted rates of the adjusting center and size are high. This implies the step size 0.1 is small. Also in the middle of Fig. 10(c), since the accepted rate of the adjusting shape is low, the step size 0.1 is too large.

For using variable step size in Fig. 11, the adjustment gets fast error decreasing and fast convergence. The step size at the 1,000*th* cycle are 0.2335, 0.0174, and 0.2356 for **c**, **A**, and *r* correspondingly.

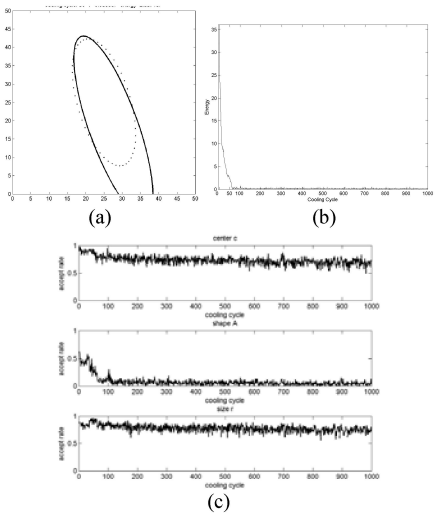

Fig. 10. Results of fixed step size − (a): after 50 cycles, (b): plot of error vs. cooling cycle. (c): accepted rate of **c**, **A**, and *r*.

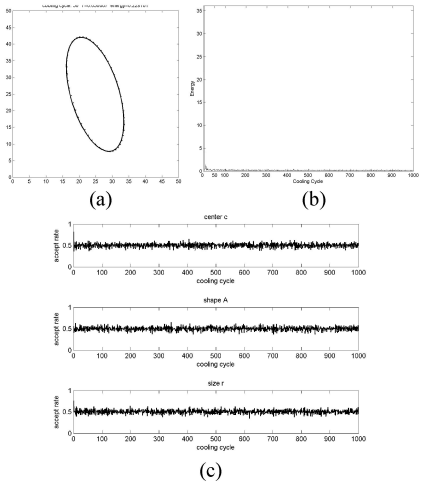

Fig. 11. Results of variable step size − (a): after 50 cycles, (b): plot of error vs. cooling cycle. (c): accepted rate of **c**, **A**, and *r*.

TABLE I
DETECTED PARAMETERS IN FIG. 8 (C) IN IMAGE SPACE 512X65

	c_x	c_y	a_{11}	a_{22}	r
Reflection wave	33.0344	55.1778	-11.5977	0.0862	48.8329
Direct wave	33.0146	12.9000	-12.3629	0.0809	5.8269

TABLE II
DETECTED PARAMETERS IN FIG. 9 (C) IN IMAGE SPACE 512X65

	c_x	c_y	a_{11}	a_{22}	r
Reflection wave	30.2009	20.2330	-12.2347	0.0817	52.3164
Direct wave	32.9537	3.1057	-12.7919	0.0782	7.5254

E. Determination of the Number of Patterns

In HTNN [6], the number of patterns was chosen by comparing the results from different number of patterns. Here we propose a method to determine the number of patterns, K, in the image. We define the detection error as

$$S = \frac{1}{N}\sum_{i=1}^{N} \min(d_1(\mathbf{x}_i), d_2(\mathbf{x}_i), ..., d_K(\mathbf{x}_i)) \quad (19)$$

where N is the number of input points. Equation (19) implies that the detection error is the average of the minimum distance from N points to their nearest patterns. Algorithm runs from pattern number K=1, 2, …, until the detection error has a minimum or no improvement. At that time, the best choice of K is determined. Fig. 12 has two circles and two ellipses and shows the result of getting K automatically. In Fig. 12(f), the detection error greatly decreases and reaches minimum when K=4. So we choose K=4. Table III lists the detection error in Fig. 12(a)-(e).

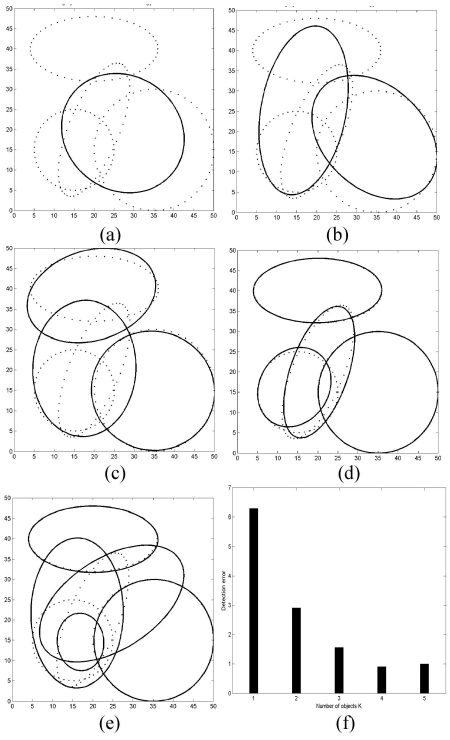

Fig. 12. Choice of K – (a) K=1, (b) K=2, (c) K=3, (d) K=4, (e) K=5, and (f) detection error of (a), (b), (c), (d), and (e).

TABLE III
DETECTION ERROR IN FIG. 12

K	1	2	3	4	5
Error	6.283951	2.907172	1.557019	0.901486	0.998616

IV. CONCLUSION AND DISCUSSION

SA algorithm detects the line, circle, ellipse, and hyperbola by finding their parameters in an unsupervised manner and global minimum fitting error related to points in an image. The formulas used in the detections of ellipse and hyperbola are different because of the properties of the pattern matrix A. The iterative adjustment requires less memory space. Also, we define the distance from point to pattern and this makes the computation feasible, especially for hyperbola. Using three steps to adjust parameters from center, shape, to size of the pattern can get fast convergence. Moreover, in order to speed the convergence, we adjust the variable learning step based on the accepted counts. Using the average minimum distance, we have proposed a method to determine the number of patterns automatically. Experimental results on the detection of line, circle, ellipse, and hyperbola in images are quite successful. The detection results of line pattern of direct wave and hyperbolic pattern of reflection wave in one-shot seismogram are good, and can improve seismic interpretations and further seismic data processing.

In the cooling schedule, the value of $T_{\max}$, N_r, and N_s are tested many times. The used value in this paper can get good performance in the experiments. Also we find the setting of $T_{\max}$ is proportional to the image size.

In the ellipse detection, matrix **A** is symmetric positive definite, so in the distance computation, we can use $d_k(\mathbf{x}_i) = \left|\sqrt{(\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{A}_k (\mathbf{x}_i - \mathbf{c}_k)} - r_k\right|$ [6]. And (3) can be just for hyperbolic detection.

In seismic application, we have no constraint on the center. However, for ideal case, the hyperbola has the center on x-axis, i.e. t=0. In simulated seismic data, we can find that the center does not lie on the x-axis, because convolution produces a shift. So preprocessing is quite critical. Wavelet and deconvolution processing may be needed in the preprocessing to improve the detection result.

REFERENCES

[1] P. V. C. Hough, "Method and Means for Recognizing Complex Patterns," U.S. Patent 3069654, 1962.

[2] R. O. Duda and P. E. Hart, "Use of Hough transform to detect lines and curves in pictures," *Commun. ACM*, vol. 15, pp. 11-15, 1972.

[3] M. M. Slotnick, *Lessons in Seismic Computing*, The Society of Exploration Geophysicists, 1959.

[4] M. B. Dobrin, *Introduction to Geophysical Prospecting*, 3rd ed., New York: McGraw-Hill, 1976.

[5] K. Y. Huang, K. S. Fu, T. H. Sheen, and S. W. Cheng, "Image processing of seismograms: (A) Hough transformation for the detection of seismic patterns; (B) Thinning processing in the seismogram" *Pattern Recognition*, vol. 18, no.6, pp.429-440, 1985.

[6] J. Basak and A. Das, "Hough transform network: A class of networks for identifying parametric structures," *IEEE Trans. Neural Networks*, vol. 13, pp. 381-392, 2002.

[7] K.Y. Huang, J. D. You, K. J. Chen, H. L. Lai, and A. J. Don , "Hough Transform neural network for seismic pattern detection," *International Joint Conference on Neural Networks*, pp. 4670-4675, 2006

[8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing", *Science* vol. 220, pp.671-680, 1983.

[9] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution and Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp.721-741, 1984.

[10] A. Corana, M. Marchesi, C. Martini, and S. Ridella, "Minimizing multimodal functions of continuous variables with the simulated annealing algorithm," *ACM Transactions on Mathematical Software*, vol. 13, pp. 262-280, 1987.

[11] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J Chem Phys.*, pp.1087–1092, 1953.