[15] T. O. Kvålseth, E. R. F. W. Crossman, and K. Y. Kum, "The effect of cost on the sampling behavior of human instrument monitors," in T. B. Sheridan and G. Johannsen, Eds., *Monitoring Behavior and Supervisory Control.* New York: Plenum, 1976.
[16] R. D. Smallwood, "Internal models of the human instrument monitor," *IEEE Trans. HFE*, vol. HFE-8, pp. 181–187, Sept. 1967.
[17] R. A. Howard, "Information value theory," *IEEE Trans. SSC*, vol. SSC-2, pp. 22–26, Aug. 1966.
[18] G. M. Jenkins and D. G. Watts, *Spectral Analysis and its Applications.* San Francisco: Holden-Day, 1968.

An Application of Relaxation Methods to Edge Reinforcement

BRUCE J. SCHACHTER, AMOS LEV, STEVEN W. ZUCKER, MEMBER, IEEE, AND AZRIEL ROSENFELD, FELLOW, IEEE

***Abstract*—An iterative scheme is used to reinforce the continuous edges detected by a differencing operation. Edges reinforce other edges that continue them and also interact with nearby nonedge points in specified ways. The slopes of the edges are also iteratively adjusted. Examples of the performance of this scheme are given.**

I. Introduction

Interest has recently arisen in a class of iterative techniques, known as "relaxation methods," that can be used to label parts of an image or scene. For a general introduction to relaxation methods and some examples of their use see [1], [2].

One of the applications in which relaxation techniques have proved useful is the detection of long, smooth curves in an image [3], [4]. Initially, line detection operators (in many orientations) are applied to the image, and their outputs are used to determine an initial level of confidence, or "probability," that each point lies on a curve in a given orientation. These probabilities are then iteratively reinforced: a point's probability of lying on a curve is reinforced by the probabilities of other points lying on curves that smoothly continue it. This reinforcement process is applied independently to every point. After a few iterations of the process, points that lie on smooth curves tend to have high probabilities of being curve points, while other points' probabilities are low.

This correspondence presents an analogous application of relaxation methods to the detection of major edges in an image. Initially, a "gradient" edge detection operator is applied to the image; this yields an edge strength (or "probability") and orientation at each point. These probabilities are then iteratively reinforced, much as in the curve case (except that edge orientations must be specified modulo 360° rather than modulo 180°, to insure that dark sides correspond to dark sides and light sides to light sides). As we shall see, this reinforcement process can be used to enhance major edges.

The general idea of the edge reinforcement process is as follows.

1) Initially, the magnitude and direction of the image gradient are computed at each point (x,y). The magnitude at (x,y), divided by the max of the magnitudes over the entire image, defines the "probability" of an edge at (x,y). [This definition is somewhat extreme; one could, alternatively, divide by the max of the magnitudes over some specified neighborhood of (x,y).] For the details of this, see Section II-A.

2) The reinforcement process defines a new edge probability at (x,y) in terms of the old ones at (x,y) and its neighbors. The computation of the new edge probability can be broken into several steps as follows.

a) Interactions between edge and nonedge probabilities: Coupling coefficients R_{ee}, R_{en}, R_{ne}, and R_{nn} are defined between the probabilities at pairs of neighboring points. The edge/edge coefficient R_{ee} depends on the edge slopes at the two points, on the slope of the line joining the points, and on the distance between them. Collinear edges reinforce, and anticollinear edges weaken one another. The particular R_{ee} used in these experiments is defined in Section II-B. Similarly, edges are weakened by nonedges collinear with them, nonedges are strengthened by edges alongside them, and nonedges are strengthened by edges near them. Definitions of the R_{en}, R_{ne}, and R_{nn} used in our experiments are given in Section II-C.

b) To compute the new edge probability resulting from these interactions, we first compute weighted sums of the edge and nonedge probability increments (see (6) and (7) in Section II-D). These sums are then normalized to lie in the range $[-1,1]$ (see (8)). They are used to update the edge and nonedge probabilities (9) and (10), and the updated values are normalized so that they sum to 1 (see (11)). For the details of this process see Section II-D.

II. Edge Reinforcement Scheme

A. Initial Edge Values

A digital gradient operation is applied to the given image f. If we denote the x and y components of the gradient by $\Delta_x f$ and $\Delta_y f$, then the magnitude and direction of the gradient are given by

$$\text{mag} = \sqrt{(\Delta_x f)^2 + (\Delta_y f)^2}$$

and

$$\theta = \tan^{-1}\left(\frac{\Delta_y f}{\Delta_x f}\right).$$

We define the "probability" of an edge at a given point (x,y) by

$$P(x,y) = \frac{\text{mag}\ (x,y)}{\max\limits_{u,v}\ \text{mag}\ (u,v)} \tag{1}$$

where the max is taken over the entire image. The probability of a nonedge at (x,y) is defined as $\bar{P}(x,y) = 1 - P(x,y)$. Clearly, $0 \le P$, $\bar{P} \le 1$ for all (x,y). In order to facilitate the reinforcement process, $P(x,y)$ was never allowed to take on the value 1; at points where mag (x,y) took on its maximum value, $P(x,y)$ was set to 0.9.

B. Edge/Edge Interaction

Let α be the edge slope at (x,y), β the edge slope at (u,v), γ the slope of the line joining (x,y) to (u,v), D the chessboard distance[1] from (x,y) to (u,v), i.e., $\max(|x-u|, |y-v|)$. Then the edge/edge reinforcement process between the points (x,y) and (u,v) has strength given by

$$R_{ee} = \cos(\alpha - \gamma)\cos(\beta - \gamma)/2^D. \tag{2}$$

To see the significance of this definition, we consider a few simple examples. In these examples, the arrows indicate the direction along the edge, with the dark side of the edge on the left.

[1] Using Euclidean distance would have weighted diagonal adjacencies more weakly than horizontal or vertical adjacencies, resulting in an iterative weakening of diagonal edges.

Manuscript received January 31, 1977. This work was supported by the National Science Foundation under Grant MCS-72-03610.

B. J. Schachter and A. Rosenfeld are with the Computer Science Center, University of Maryland, College Park, MD 20742.
A. Lev is with the Computer Science Department, Ford Motor Company, Dearborn, MI.
S. W. Zucker is with the Department of Electrical Engineering, McGill University, Montreal, PQ, Canada.

	Case	α	β	γ	$\cos(\alpha - \gamma)\cos(\beta - \gamma)/2^D$
a)	↑↑	90	90	0	0
b)	↑↓	90	270	0	0
c)	↑ ↑	90	90	90	$1/2^D$
d)	↑ ↓	90	270	90	$-1/2^D$
e)	↑ →	90	0	0	0
f)	↑ ←	90	180	0	0

We see from these examples that parallel and perpendicular edges have no effect on one another, that collinear edges reinforce each other to a degree that decreases with distance, and that anticollinear edges weaken each other, decreasingly with distance.

Other edge/edge reinforcement schemes can easily be devised. For example, in case a) one might want to strengthen the stronger of the two edges and weaken the weaker one [5]; this process would have the effect of thinning thick edges. In case e), one might want the edges to weaken one another, and in case f) to reinforce one another, since in the former case the light side of one edge corresponds to the dark side of the other, while in the latter case the light sides coincide. Also, the reinforcement weights need not vary as the cosines of the angles, nor fall off exponentially with distance. However, in the present study, only the simple reinforcement scheme defined by (2) was used.

C. Interactions Involving Nonedges

Besides the edge/edge interaction described in Section II-B, which occurs between the edge probabilities P, there are also interactions involving the nonedge probabilities $\bar{P}$.

1) The edge probability at (x,y) is weakened by the nonedge probability at (u,v) to the degree R_{en} defined by

$$R_{en} = \min\,[0, -\cos(2\alpha - 2\gamma)/2^D]. \tag{3}$$

2) The nonedge probability at (x,y) is affected by the edge probability at (u,v) to the degree R_{ne} defined by

$$R_{ne} = (1 - \cos(2\beta - 2\gamma))/2^{D+1} \tag{4}$$

3) The nonedge probabilities at (x,y) and (u,v) reinforce each other to the degree R_{nn} defined by

$$R_{nn} = 1/2^D. \tag{5}$$

The R_{nn} reinforcement process is easy to understand; nonedge probabilities are reinforced by other nearby nonedge probabilities. To see the significance of the R_{en} process, we again consider some simple cases (here the arrows indicate edge points, as in Section II-B, and the dots indicate nonedge points).

	Case	α	γ	$-\cos(2\alpha - 2\gamma)/2^D$	R_{en}
a)	↑ ·	90	0	$+1/2^D$	0
b)	↓ ·	270	0	$+1/2^D$	0
c)	↑ ·	90	90	$-1/2^D$	$-1/2^D$
d)	· ↑	90	90	$-1/2^D$	$-1/2^D$

Thus nonedge points collinear with edge points weaken them, whereas nonedge points alongside edge points have no effect on them.

Similarly, the R_{ne} process is illustrated by the following cases.

	Case	β	γ	$(1 - \cos(2\beta - 2\gamma))/2^{D+1}$
a)	↑ ·	90	0	$1/2^D$
b)	↑ ·	90	90	0

Here we see that edge points alongside nonedge points strengthen them, while edge points collinear with nonedge points have no effect on them. Other nonedge interaction schemes could have been devised; for example, the R_{en} formula could have been used for R_{ne} too. However, in this correspondence we have studied only the schemes defined by (3)–(5).

D. Combined Reinforcement Process

For each point (x,y), the net effect of its neighboring points on its edge probability $P(x,y)$ and nonedge probability $\bar{P}(x,y) = 1 - P(x,y)$ is computed as follows. Let

$$Q(x,y) = \sum_{\substack{(u,v) \\ \neq (x,y)}} C_1 P(u,v) R_{ee}((x,y),(u,v)) + C_2 \bar{P}(u,v) R_{en}((x,y),(u,v)) \tag{6}$$

$$\bar{Q}(x,y) = \sum_{\substack{(u,v) \\ \neq (x,y)}} C_3 P(u,v) R_{ne}((x,y),(u,v)) + C_4 \bar{P}(u,v) R_{nn}((x,y),(u,v)). \tag{7}$$

Here C_1, C_2, C_3, C_4 are constants whose sum is taken to be one. Let

$$\hat{Q} = \frac{Q}{|Q| + |\bar{Q}|} \qquad \hat{\bar{Q}} = \frac{\bar{Q}}{|Q| + |\bar{Q}|}. \tag{8}$$

Finally, let

$$P' = P[1 + \hat{Q}] \tag{9}$$

$$\bar{P}' = \bar{P}[1 + \hat{\bar{Q}}] \tag{10}$$

and

$$P^{\text{new}} = \frac{P'}{P' + \bar{P}'}. \tag{11}$$

This process is then iterated, with P^{new} replacing P, and $1 - P^{\text{new}}$ $(= \bar{P}'/(P' + \bar{P}'))$ replacing $\bar{P}$.

In addition to computing new edge and nonedge probabilities for each point at each iteration, we also recompute the estimated edge direction at each point. This is done as follows. Let

$$\hat{\Delta}_x(x,y) = WP(x,y)\cos(\theta(x,y)) + \sum_{\substack{(u,v) \\ \neq (x,y)}} P(u,v) R_{ee}((x,y),(u,v)) \cos(\theta(u,v))$$

$$\hat{\Delta}_y(x,y) = WP(x,y)\sin(\theta(x,y)) + \sum_{\substack{(u,v) \\ \neq (x,y)}} P(u,v) R_{ee}((x,y),(u,v)) \sin(\theta(u,v)).$$

Then

$$\theta^{\text{new}}(x,y) = \tan^{-1}(\hat{\Delta}_y(x,y)/\hat{\Delta}_x(x,y)).$$

Note that for large values of the constant W, θ^{new} is close to θ; while for small values, it is strongly influenced by the neighboring θ's.

III. Experimental Results

Fig. 1 shows a portion of a LANDSAT image of Monterey, CA. Fig. 2 shows approximate gradient magnitudes for Fig. 1; the magnitude at a point was computed as the max (rather than square root of sum of squares) of the first differences in the x and y directions. Fig. 3(a) shows the gradient values displayed as an

Fig. 1. Input image.

Fig. 2. Gradient magnitudes for Fig. 1.

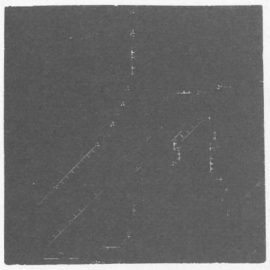
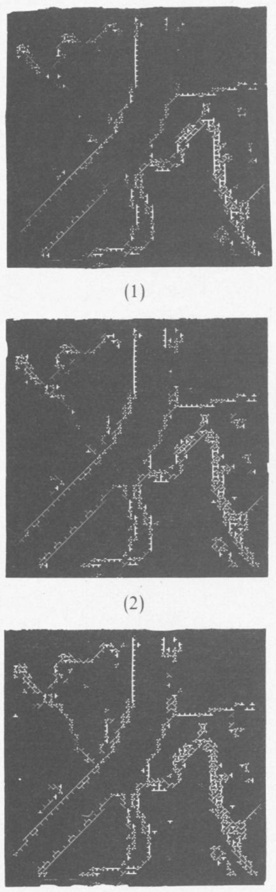


Fig. 3. Gradient magnitudes and directions for Fig. 1, displayed as intensities and orientations of T-shaped symbols.

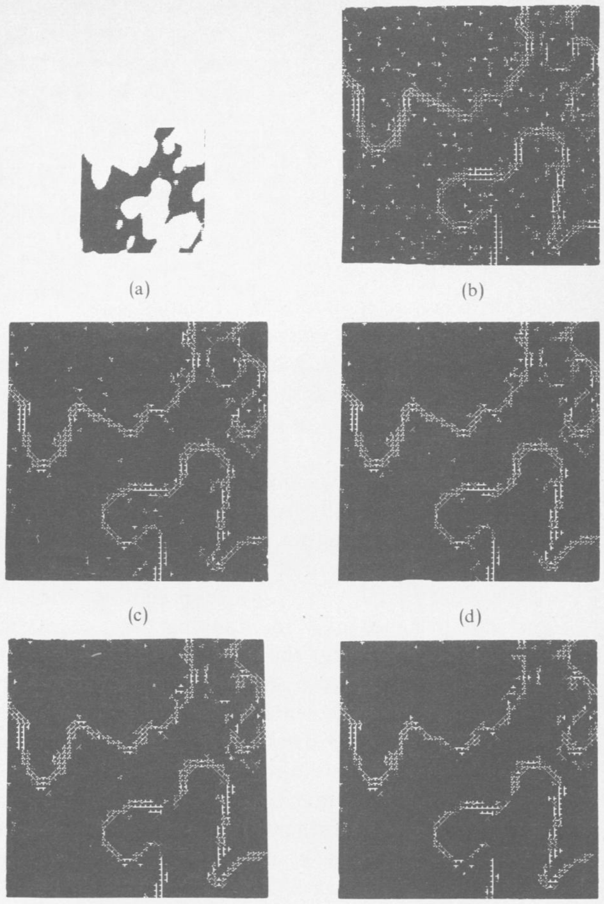


(1)

(2)

(3)

Fig. 4. Three iterations of edge reinforcement process applied to Fig. 3.



(a) (b)

(c) (d)

(e) (f)

Fig.5. Four iterations of edge reinforcement process applied to noisy edge data. (a) Original picture (leaves). (b) Edge output for (a) with noise added. (c)–(f) Four iterations.

array of T-shaped symbols, where the intensity of a symbol corresponds to the gradient magnitude, and its slope (quantized to a multiple of $22\frac{1}{2}°$) indicates the gradient direction, with the leg of the T pointing toward the light side of the edge.

Fig. 4 shows the results of three iterations of the edge reinforcement process, as described in Section II, applied to Fig. 3. In this implementation, the points outside a 5-by-5 square centered at (x,y) were ignored; thus only 24 (u,v)'s influenced a given (x,y). The values of the constants of Section II-D were as follows:

$$C_1 = 0.866 \qquad C_2 = 0.124 \qquad C_3 = C_4 = 0.005 \qquad W = 3.$$

Further iterations yielded little change from the result shown in the last part of Fig. 4.

The results of the iteration process are somewhat sensitive to the choice of the C's. For example, if C_1 is too large, edges will thicken and will be extended into nonedge points; while if C_4 is too large, gaps will appear at weak spots in the edges and at sharp angles. Perhaps different C's should be used at each point, depending on the level of edge activity in the vicinity of that point, and the values of the C's could also vary from iteration to iteration. These ideas will not be pursued further here, but they should be seriously considered in future implementations.

The iteration scheme has considerable noise cleaning power, as shown by the example in Fig. 5. (This example used a more elaborate edge detection scheme developed in connection with a relaxation-based noise smoothing process [6].) In Fig. 5, Gaussian noise was added to the edge detector output,[2] but a few iterations of the reinforcement process delete nearly all of this noise. These experiments used $C_1 = 0.706$, $C_2 = 0.176$, $C_3 = C_4 = 0.059$; note that because of the smaller value of C_1, some of the weaker, higher curvature edges (e.g., at the upper right) have begun to disappear by the fourth iteration.

IV. Conclusions

The experiments reported here confirm that iterative reinforcement schemes can be used to "enhance" edge points belonging to extended edges, while weakening or deleting noisy edge points. The iterative process used was not designed to handle sharp angles on edges; in spite of this, its performance seems reasonably good.

Raw edge detector output is often quite noisy, so that it is difficult to obtain good results from curve following or fitting procedures applied to such output. In such a situation, iterative procedures can be used to "clean up" the edge output and make it more tractable to further processing. More specialized procedures of this type, e.g., designed only for thinning thick edges, have already been proposed [5], but the present work shows that a general reinforcement process can also be carried out iteratively.[3] Such processes could be implemented very simply and efficiently on parallel array-processing hardware. It is expected that they will come into increaing use over the coming years.

It should be emphasized that the results reported here are illustrative of a wide variety of schemes that could have been used. The reinforcement processes could have been defined and combined in many different ways. The results obtained will certainly depend on how this is done. The question of how to choose the reinforcement coefficients for a given image needs further investigation. Also, no claim is made here about the convergence properties of the process (see, however, [1] and [7]). However, it does appear that a few iterations of the process tend to weaken or eliminate noise while preserving long edges.

Acknowledgment

The help of Ms. Shelly Rowe in preparing this correspondence is gratefully acknowledged.

References

[1] A. Rosenfeld, R. A. Hummel, and S. W. Zucker, "Scene labeling by relaxation operations," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, pp. 420–433, June 1976.

[2] S. W. Zucker, "Relaxation labeling and the reduction of local ambiguities," in *Proc. 3rd Int. Joint Conf. Pattern Recognition*, San Diego, CA, pp. 852–861, Nov. 1976.

[3] S. W. Zucker, R. A. Hummel, and A. Rosenfeld, "An application of relaxation labeling to line and curve enhancement," *IEEE Trans. Comput.*, vol. C-26, pp. 394–403, Apr. 1977.

[4] G. J. VanderBrug, "Experiments in iterative enhancement of linear features," *Computer Graphics Image Processing*, vol. 6, pp. 25–42, Feb. 1977.

[5] R. B. Eberlein, "An iterative gradient edge detection algorithm," *ibid.* 5, pp. 245–253, 1976.

[6] A. Lev, S. W. Zucker, and A. Rosenfeld, "Iterative enhancement of noisy images," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, pp. 435–442, June 1977.

[7] S. W. Zucker, E. V. Krishnamurthy, and R. L. Haar, "Relaxation processes for scene labelling: Convergence, speed, and stability," *IEEE Trans. Syst., Man, Cybern.*, submitted.

[8] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*. New York: Academic, 1976, pp. 277, 313.

[2] This noise had a standard deviation of 200 (on a gray scale of 0 to 63); but if the signal plus noise fell outside the range [0,63], the noise was not added. The edge slopes of the points to which noise was added were chosen randomly.

[3] An alternative approach to edge reinforcement is to (iteratively) apply curve detectors, in all orientations, to the edge detector output [8]. This can be regarded as a simplification of the present scheme, in which edge orientations are ignored.

Pattern Classification by a Learning Algorithm Similar to Hebb's Modifiable Synapse

TAKASHI NAGANO

***Abstract*—The capability of a learning algorithm like Hebb's modifiable synapse (called "forced learning procedure" hereafter) is discussed. The necessary and sufficient condition for a multiclass pattern classifier with the forsed learning procedure to classify input patterns correctly is introduced. Differences between the forced learning procedure and the error correction procedure are also discussed. Although the error correction procedure gives better pattern classification than the forced learning procedure, hardware systems with the forced learning procedure are more simply built up than those with the error correction procedure. It is also concluded that the forced learning procedure is a more plausible manner of changing synaptic efficacies of nervous networks.**

I. Introduction

Learning machines have been studied from various viewpoints for almost twenty years since Rosenblatt proposed Perceptron [1]. Functions of these machines are roughly classified into two groups: one is the pattern classification function [1]–[5], and the other is the association function [6]–[9]. These functions are produced by organizing learning machines according to certain learning algorithms. Though various types of learning algorithms for learning systems with a teacher have been proposed, they can be grouped into one class called error correction procedures. Much work has been done on the pattern classification capability [10] and convergence of learning process of this type of algorithm [5], [11]. They have been proved to be the best algorithm theoretically in the sense that they can classify two sets of input patterns which are linearly separable, or that they can produce the optimum decision hyperplane under a certain criterion even when two pattern sets are not linearly separable.

These algorithms, however, do not seem to be the best when the difficulty of the realization of their hardware systems is taken into account. This correspondence reports the pattern classification capability of a learning algorithm which can be built up as a hardware system more easily than the error correction procedure. It is also shown that this algorithm is plausible as a way of changing synaptic efficacies (a kind of coupling coefficient between two neurons) of the nervous system, since it is similar to the Hebb's modifiable synapse [12].

II. Error Correction Procedure

The following notation will be used in this paper:

U_α $(\alpha = 1, 2, \cdots, m)$	computational unit (Fig. 1),
θ	threshold of U_α $(\theta > 0)$,
$X = (x_1, x_2, \cdots, x_n)$	input pattern,
C_α, $(\alpha = 1, 2, \cdots, m)$	class of input patterns,
$X_\alpha = (x_{1\alpha}, x_{2\alpha}, \cdots, x_{n\alpha})$	input pattern in C_α,
$W_\alpha = (w_{1\alpha}, w_{2\alpha}, \cdots, w_{n\alpha})$	weighting function of U_α (see Fig. 1),
W_α^*	weighting function after learning,
T_α	element supplying a desired output signal to U_α,
d_α (= 1 or 0)	desired output of U_α (the output of T_α),
y_α	summation of inputs of U_α which is defined by

$$y_\alpha \triangleq (W_\alpha, X) = \sum_{i=1}^{n} w_{i\alpha} x_i, \quad (1)$$

Manuscript received September 11, 1975; revised November 1, 1976.

The author is with the Information Science Division, Electrotechnical Laboratory, Ministry of International Trade and Industry, Tokyo 188, Japan.