IEEE/RSJ International Workshop on Intelligent
Robots and Systems IROS '91, Nov. 3-5, 1991,
Osaka, Japan. IEEE Cat. No. 91TH0375-6

FUZZY RULE-BASED BOUNDARY ENHANCEMENT ALGORITHM FOR NOISY IMAGES

J.S.Kim*, H.S.Cho* and S.K.Kim**

* Department of Production Engineering
Korea Advanced Institute of Science & Technology
P.O. Box 150, Chongryangri, Seoul, Korea

** Production Technology Division, Samsung Electronics Co.Ltd.
Suweon, Korea

ABSTRACT

This paper presents a new edge relaxation algorithm that enhances the noisy boundary informations in image. The proposed algorithm employs relaxation process that reduces or even eliminates ambiguities of derivative operator response via contextual informations. The contextual informations are the neighborhood patterns of a central edge which are estimated using fuzzy pattern matching technique. The algorithm is developed on crack edges. Experimental results show that the proposed scheme is effective even for noisy images or low contrast images.

1. INTRODUCTION

Basically, an idea underlying most edge detection techniques is to convolve an image with a local derivative operator that evaluates the derivative of image intensity within local neighborhood. A variety of derivative operators have been defined in the literature (refer to [1-3] for review). Ideally, derivative operator should strongly respond to only pixels lying on the object boundary, changes in surface orientation or material properties. In practice, noise contained in input scene or texture internal to a region makes the relationship between a derivative operator response and the existence of an edge element ambiguous. That is, strong response may occur when no boundary segment is present, and vice versa. To overcome the problem, decisions regarding the existence of boundary segments should not be made purely at the local level, but instead should be delayed until contextual information can be brought to bear on the decision, thus reducing ambiguities sufficiently.

An ambiguity-reduction operation, which is called a relaxation process, was formulated by Rosenfeld, Hummel and Zucker[4]. They showed that ambiguities can be reduced, or even eliminated by using contextual information. Since the formulation of this relaxation process, many different relaxation techniques have been applied to various areas, such as scene labeling[4], line detection[5], region merging[6] and template matching[7].

Zucker, Hummel and Rosenfeld[5] applied the relaxation process to detect smooth lines and curves in noisy, real world image. the algorithm has a drawback in that the proportionality relationships between the compatibility coefficients, which are defined between the labels, must be dramatically changed to favor the specific circumstances in particular applications. Hanson and Riseman[6] proposed an edge relaxation technique that attempts to overcome the above limitation. They generalized the notion of a label to that of a pattern defined over the neighborhood, not to that of a neighboring edge. The approach gives a significant improvement over earlier approaches. Prager[8] developed an alternative edge relaxation algorithm that is a modified form of Hanson and Riseman's scheme. It has a difficulty in that the updating constant k should be well tuned considering that a large value for k gives fast convergence, but does not permit information to propagate far, and vice versa. Leung and Li[9] presented several parallel algorithms for edge relaxation on array processors with different numbers of processing elements connected by a mesh or hypercube network. The schemes make modifications to Prager's scheme so that no multiplications are employed and only integer operations are required. It is a common drawback in the aforementioned techniques that the classification of neighborhood pattern is performed through the approaches based on probabilistic theory or two-valued logic, even if the notion of an neighborhood pattern is not defined precisely.

This paper presents a new edge relaxation algorithm that enhances boundary informations. The motives of the proposed algorithm are in the classification of neighborhood pattern using fuzzy technique, and its application to edge enhancement. There are two reasons for using fuzzy technique in estimating the neighborhood patterns. First, it provides a convenient mechanism for roughly representing the concept of an object using natural language. Typically each vertex type can be described by using imprecise linguistic terms, even if it is impossible to accurately characterize the vertex type. Secondly, it is also computationally simple because it requires only simple operations, maximum, minimum and complement.

To investigate the validity of the proposed algorithm, a series of experiments are conducted for the images contaminated with noise. Section 2 provides some considerations in edge representation for relaxation. Section 3 proposes the fuzzy rule-based relaxation algorithm. Section 4 exhibits the experimental results and discusses. Finally, a conclusion is presented in Section 5.

2. EDGE REPRESENTATION

The standard technique for edge detection is differentiation of intensity in local area that is to convolve edge masks with the image. The resulting outputs, edge strengths and/or directions, are usually associated with the central pixel in the local window. An edge represented in this way exhibits the following characteristics (Consider the Kirsch derivative operator with eight possible orientations and placements of an edge as standard operator):

1) An edge passes through a pixel.
2) An edge has an ambiguous location and orientation on array of pixels.
3) There are multiple indications of a single local edge.
4) Competition between edges causes an interesting edge to be suppressed by another edge in the vicinity.

These complexities are overcome by the choice of interpixel representation of horizontal and vertical edge, which is called crack edge, depicted in Fig.1 (See [6] for more detailed discussions). This representation has been used in some early works on edge relaxation[8] and region growing[6]. In this representation, the possible locations of edges are clearly defined and each edge has a unique position and orientation with respect to the pixels from which they are obtained. Thus, this greatly facilitates further processing, such as relaxation and curve fitting.
To generate the crack edge, let S(i,j) denote the strength of crack edge at position (i,j). The strength of an edge is determined by computing the absolute difference in average intensity levels between two local areas sharing each other. Then,

$$S(i,j+1) = |A(i,j) - A(i,j+2)|$$
and
$$S(i+1,j) = |A(i,j) - A(i+2,j)|$$

are the strength of a horizontal edge and a vertical edge respectively. Herein, G(i,j) denotes a grey level of intensity at position (i,j), and average intensity level A(i,j) is defined as follows:

$$A(i,j) = \{G(i,j)+G(i+1,j)+G(i,j+1)+G(i+1,j+1)\}/4.$$

Fig.2 depicts the detailed procedure for generating crack edges. The proposed edge relaxation algorithm, which will be described in next section, operates on this crack edge.

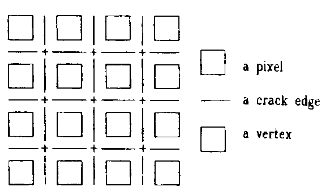

Fig.1. Representation of the crack edges (horizontal line and vertical line between pixels denote crack edges : horizontal edge and vertical edge respectively).

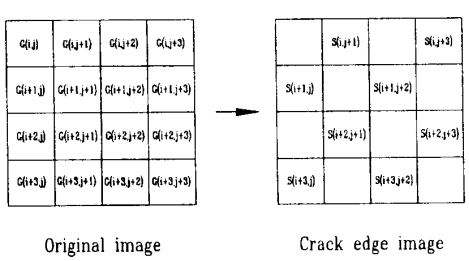

Fig.2. Generation of the crack edges

3 EDGE RELAXATION ALGORITHM

3.1. A brief review of relaxation process

A brief review of key ideas of relaxation processes will be given in this subsection (refer to [4,6,10] for more detailed discussions). Relaxation operates on objects with labels attached to them. Some labels are strengthened and some are weakened or eliminated through the use of approximately defined compatibility relationships between labels. Let $a = \{a_1, a_2, \cdots, a_n\}$ be a set of objects. Let $\Lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_m\}$ be a set of labels which indicate possible interpretations for these objects. A probability $P_i(\lambda_k)$ is attached to each label where λ_k is the correct label for the object a_i. The relaxation process updates the estimated probabilities in an iterative manner. $P_i(\lambda_k)$ will be increased (or decreased) by label λ_l if the labels are compatible (or incompatible), where the effect of this change is weighted by $P_j(\lambda_l)$. Compatibility is defined in terms of a function r_{ij}:

$$r_{ij}(\lambda_k,\lambda_l) > 0 \ \text{ if } \lambda_k \text{ and } \lambda_l \text{ are compatible},$$
$$r_{ij}(\lambda_k,\lambda_l) < 0 \ \text{ if } \lambda_k \text{ and } \lambda_l \text{ are incompatible},$$
$$r_{ij}(\lambda_k,\lambda_l) = 0 \ \text{ if } \lambda_k \text{ and } \lambda_l \text{ are independent}.$$

The updating process can now be expressed in terms of these compatibility functions. Let $Q_i^t(\lambda_k)$ represent the correction applied to $P_i^t(\lambda_k)$ in the (t+1)st iteration to obtain $P_i^{t+1}(\lambda_k)$:

$$P_i^{t+1}(\lambda_k) = \frac{P_i^t(\lambda_k)[1+Q_i^t(\lambda_k)]}{\sum_{k=1}^{m}[P_i^t(\lambda_k)[1+Q_i^t(\lambda_k)]]}$$

where the correction is defined as follows:

$$Q_i^t(\lambda_k) = \sum_j d_{ij}[\sum_{l=1}^{m} r_{ij}(\lambda_k,\lambda_l)P_j^t(\lambda_l)].$$

The coefficients d_{ij} weight the total influence that object a_j can have on a_i, subject to $\Sigma d_{ij} = 1$. The relaxation process operates in this way until limiting values are obtained for the label probabilities.
Now we will look at the application of this relaxation process to edge enhancement. Let us consider the crack edge representation. The relaxation is very much simplified as a result of this representation. There will be only two labels at each edge position, "edge" and "noedge". The two labels may be regarded as competing at each edge position during relaxation. To determine the probability of both labels, only a single probability, P_i(edge), is necessary since P_i(noedge) = 1 − P_i(edge). Now let P_i^t represent P_i^t(edge). Then we can derive the following equation for computing P_i^{t+1}:

$$P_i^{t+1} = \frac{P_i^t[1+Q_i^t]}{1+P_i^tQ_i^t}$$

where Q_i^t is the correction for an edge at the position i.

3.2. The proposed edge relaxation algorithm

We propose a new edge relaxation algorithm which utilizes fuzzy logic for estimating the neighborhood patterns. This algorithm was designed to operate on crack edge, but can be easily extended to some different types of edge representation. First, the edge strengths are normalized by the maximum value over an entire image. Then, the edge strengths are always in the range [0,1]. We regard the strength of edge at position (i,j) as the probability or confidence P_{ij} that the edge exists at the position. The overall strategy is to update the probability, or the edge strength, in an iterative manner based on the recognition of some local neighborhood patterns. Fig.3 shows the overall configuration of the proposed scheme. The smallest meaningful neighborhood for an edge is shown in Fig.4, where e denotes the strength of a central edge, and a,b,c,f,g,h,u,v denote the strengths of the neighboring edges, respectively. We will generate the algorithm on the basis of the neighborhood defined, as shown in the figure.

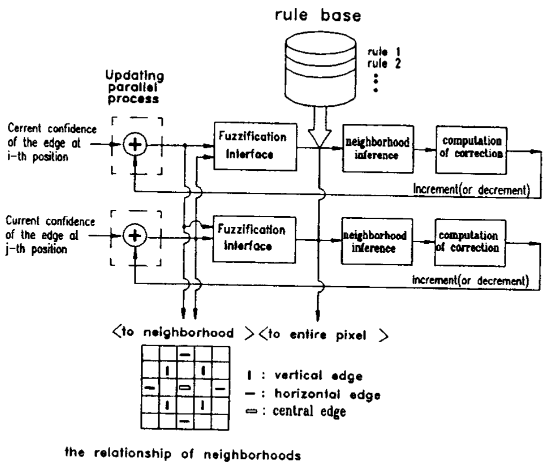

Fig.3. Overall configuration of the proposed scheme

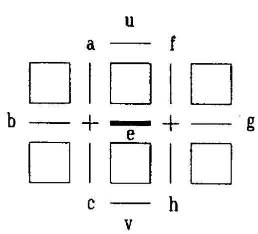

Fig.4. Local neighborhood (A thick line denotes a central edge, and its strength is represented by e.)

Definition of vertex types

Every edge position has two endpoints, or vertices at which the edge could continue. Each edge endpoint will be classified as one of four vertex types, as shown in Fig.5. The vertex type is denoted by an integer p in the range 0–3, representing the number of strong neighboring edges. A vertex type is considered equivalent no matter which of the three possible edge position that they take.

Estimation of vertex type based on fuzzy technique

The vertex types of the endpoints of the edge position under examination determine how the edge strength is to be updated. Each edge is associated with a numerical value

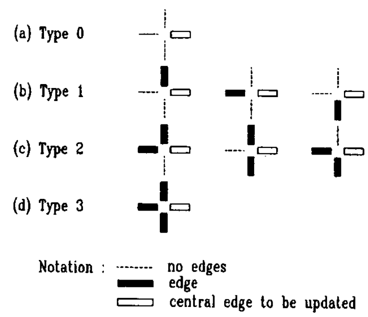

Fig.5. Vertex types of central edge (Three cases in (b) and (c) are considered equivalence classes, respectively.)

in the range of [0,1], representing the probabilities of the presence of an edge. The vertex types would be determined as a function of the probabilities of three edges to either side. However, there does not a universal formulation capable of precisely classifying the vertex types since the notion of vertex type dose not permit a precise definition. Therefore, the conceptual structure of the theory of fuzzy sets may well provide a more natural formulation for the vertex type classification than the traditional approach based on two–valued logic. This is a motive from which we utilize the fuzzy logic as a technique for estimating the confidence of the vertex being each vertex type. In particular, we employ a fuzzy pattern matching technique.

Suppose one of the vertices of a central edge has three neighboring edges with its strengths a,b and c in Fig.4. Perpendicular continuation is treated equivalent to straightline continuation. Therefore, we can assume that $a \geq b \geq c$.

First, in order to account for the natural imprecision inherent to the edge strengths, linguistic representations are used for the edge strengths. In a word, we consider the three edge strengths a,b,c as linguistic variables. Note that the three edge strengths are normalized by the following scaling factor m before the fuzzification of them:

$$m = MAX(a,q)$$

where $MAX(\cdot)$ is a maximum value of the variables inside the parenthesis and q is a lower bound for m. This operation will anchor m to some minimum value q when all incident edges have very low strengths. Each linguistic variable is assigned fuzzy membership functions which are fuzzy subsets. Two membership functions of these linguistic variables are defined as depicted in Fig.6. All the membership functions are defined in the universe of discourse ranging over [0,1], which is assumed to be continuous. They are named as "BIG" and "SMALL", respectively, to denote the notion of edge strength.

Second, the characteristic of each vertex type is described using the predefined sets of linguistic variables. The rules used to describe each vertex type in our problem are as follows:

Rule1 : if (a is SMALL and b is SMALL and c is SMALL)
then (class of vertex is type 0)

Rule2 : if (a is BIG and b is SMALL and c is SMALL)
then (class of vertex is type 1)

Rule3 : if (a is BIG and b is BIG and c is SMALL)
then (class of vertex is type 2)

Rule4 : if (a is BIG and b is BIG and c is BIG)
then (class of vertex is type 3)

Finally, we compute the degrees to which the input pattern of a vertex is similar to the standard patterns about each vertex. We will introduce $\mu_s(\cdot)$ and $\mu_b(\cdot)$ to represent the degree of membership of the linguistic variables inside the parentheses to the membership functions "SMALL" and "BIG", respectively. If we let D_i denote the degree of similarity between the input pattern of vertex and the standard pattern about vertex type i, then the D_i's are estimated by the above four rules, respectively, in the following manner:

$$D_0 = \mathrm{MIN}(\mu_s(a),\mu_s(b),\mu_s(c))$$
$$D_1 = \mathrm{MIN}(\mu_b(a),\mu_s(b),\mu_s(c))$$
$$D_2 = \mathrm{MIN}(\mu_b(a),\mu_b(b),\mu_s(c))$$
$$D_3 = \mathrm{MIN}(\mu_b(a),\mu_b(b),\mu_b(c))$$

where $\mathrm{MIN}(\cdot)$ is a minimum vale of the variables in the parenthesis. The same procedures are employed for computing the d_i's of the other vertex.

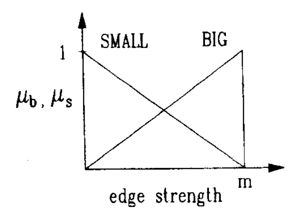

Fig.6. Membership functions for strengths of neighboring edges (a,b,c,f,g and h)

Computation of contextual patterns

Since the goal of our study is to extract continuous boundary segments, the patterns should focus on those conditions which support or inhibit the assumptions that the central edge is a part of a global boundary. The probability of a central edge will be updated via contextual informations which are represented by neighborhood patterns. Let us use the notation i–j to denote the neighborhood pattern where the vertex types at both endpoints of a central edge are i and j, respectively.
As a matter of convenience i–j and j–i will be considered as an equivalence class. Then, only cases 0–0 through 3–3 shown in Fig.7 are the possible neighborhood patterns. If we use the notation P_l to represent the confidence of an i–j vertex pair where index l denotes the serial number of i–j vertex pair, then P_l is computed in the following manner:

$$P_l = \mathrm{MIN}(D_i,D_j) \qquad \text{for } l = 1,\cdots,10.$$

Note that $i \leq j$ when the index i is changed from 0 to 3.

Updating of edge probabilities

Now, it only remains to update edge probabilities through the modified relaxation process. The compatibility coefficients are between the labels of l^{th} neighborhood pattern and the labels of central edge. Thus, r_{ij} is replaced by a weight w_l which determines the effect that the pattern l should have on central edge. A w_l is assigned a value ranging over [–1,1] determined according to the edge semantics, as shown in Fig.7. Fig.7(a) should be inhibited, i.e., in the absence of any additional information, the probability of a central edge should be reduced. Fig.7(b) and Fig.7(c) should be also inhibited so that unnecessary spurious lines do not grow out. In Fig.7(d) no updating should take place because it is dependent on a wider context. It is clear that Figs.7(e)–(g) support e. The clearest case is for 1–1 case, where e is necessary for good line continuation. The last three Figs.7(h)–(j) should not be inhibited or supported because the presence or the absence of e does not affect good line continuation. For supporting cases w_l should approach 1, for inhibiting cases w_l should approach –1, and for uncertain case w_l should be near 0. Now total contribution, or the correction, Q is defined as a linearly weighted sum of the confidence of the n patterns:

$$Q^t = \sum_{l=1}^{n} Q_l^t = \sum_{l=1}^{n} w_l P_l^t$$

Finally, we obtain a new probability of a central edge, P^{t+1}, in the following manner:

$$P^{t+1} = \frac{P^t[1+Q^t]}{1+P^tQ^t}$$

This approach could be generalized to larger neighborhood. There is one more condition that must be considered: parallel edge suppression. Wide gradients give rise to multiple parallel indications of the same edge. A simple technique for eliminating these unwanted edges is a nonmaximal suppression: All but strongest are eliminated. The relaxation is performed after this parallel edge suppression is finished.
In summary, the proposed edge relaxation algorithm can be described as follows:

step 1) Obtain initial edge strengths.
step 2) t = 1.
step 3) Compute confidence of each vertex type based on strengths of neighboring edges.
step 4) Compute confidence of each neighborhood pattern.
step 5) Update the edge strengths through relaxation process.
step 6) If a maximum number of iterations is reached, or all edge strengths converge to 0 or 1, then stop. Otherwise, increment t to t+1 and go to step 3).

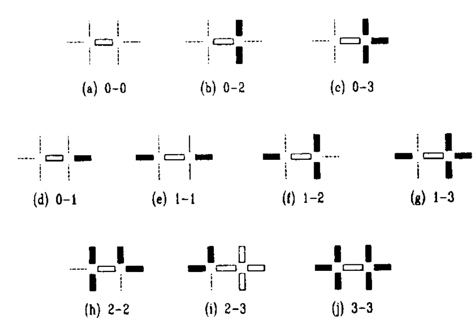

Fig.7. Neighborhood patterns

4. EXPERIMENTS

4.1 Experimental setup

The experimental system comprises three units; a camera set, an image frame grabber and a computer system. For acquisition of images, we have used a CCD TV camera with a resolution of 764 × 491 interlaced pixels. A lense system with focal length of 16*mm* has been used. The video signal from the CCD TV camera is sampled into 512 × 512 pixels and digitized into 8–bit number(256 grey levels) using the image frame grabber FG–100–1024, which is a single–board image processor. An IBM–AT 386 PC has been used for reading the image data stored in the picture memory of the frame grabber, performing image operations, analyzing the processed informations and displaying the results. The proposed algorithm has been implemented in C–language and linked with ITEX 100, which is a library of image processing subroutines written for use with the frame grabber FS–100–1024.

4.2 Experimental results and discussions

The test is performed on the images with grey level intensities and 128 × 128 pixels shown in Fig.8. Fig.8(a) shows the original image used for evaluating the capability of the proposed algorithm. Figs.8(b) and (c) represent the image embedded in a white gaussian noise with zero mean. In (b) and (c) of the figure, the signal to noise ratio(SNR), defined as the ratio of the amplitude of the step to the standard deviation of the noise at each pixel, are 0.2 and 0.4, respectively. The results of operation of the proposed algorithm on the noisy images are represented in Figs.9 and 10. Fig.9 represents the enhancment results of Fig.8(b) while Fig.10 represents those of Fig.8(c). Shown in Fig.9(a) and Fig.10(a) is the crack edge image differentiated using 2 × 4 operator described in section 2 and thinned by non–maximal suppression. Herein, the brightness of a pixel encodes the strength of the edge which has been normalized to a zero–one range by the maximum strength edge over the entire image. The results after iterations, several are shown in (b)–(d) of the figures, respectively. The results show that the added white gaussian noise was drastically reduced or even perfectly eliminated, while preserving the useful boundary information.

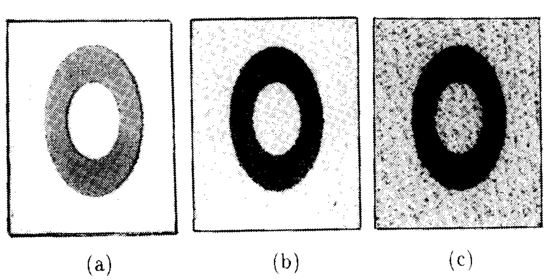

Fig.8. Test images. (a) original image. (b) test image added in white gaussian noise with SNR = 0.2. (c) test image added in white gaussian noise with SNR = 0.4.

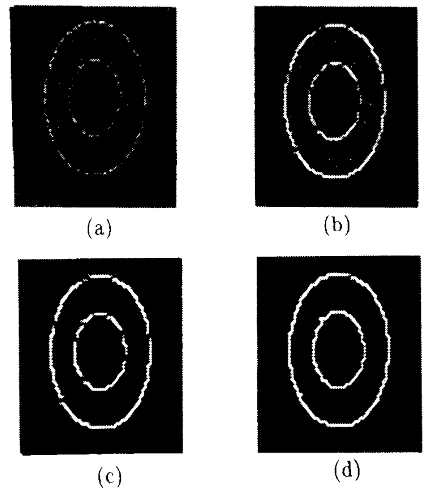

Fig.9. Results of the relaxation on the test image shown in Fig.8(b). (a) thinned and normalized edge image. (b) result after first iteration. (c) result after 3rd iteration. (d) result after 7th iteration.

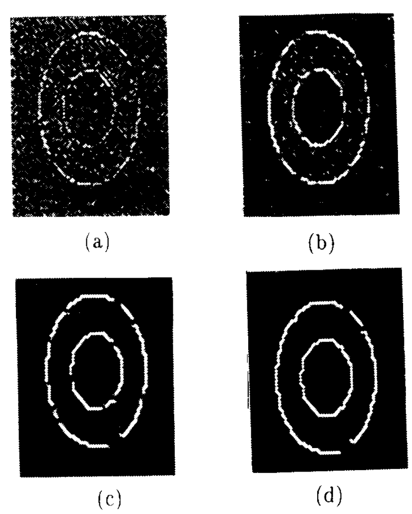

Fig.10. Results of the relaxation on the test image shown in Fig.8(c). (a) thinned and normalized edge image. (b) result after first iteration. (c) result after 3rd iteration. (d) result after 7rd iteration.

5. CONCLUDING REMARKS

This paper describes an edge enhancing method that utilizes relaxation process and a fuzzy logic–based decision making scheme to estimate neighborhood patterns of an edge. The basic motivation behind this method lies in the premise that ambiguities of edges can be reduced using contextual information and the notion of fuzzy sets can well

characterize the neighborhood patterns, which are used as contextual information in our study. In computing neighborhood patterns, the fuzzy rules replace the mathematical formulas based on two–valued logic.
The proposed algorithm has shown the ability to eliminate the false edge resulted in derivative operator. In addition, the algorithm is capable of detecting the weak edge on the low contrast boundaries and bridging the small gap between pixels on boundaries.

REFERENCES

[1] L.S.Davis, "A survey of edge detection techniques," Computer Graphics and Image Processing, Vol.4, pp.248–270, 1975.

[2] D.H.Ballard and C.M.Brown, Computer Vision, Prentice–Hall,Inc., 1982.

[3] V.S.Nalwa and T.O Binford, "On detecting edges," IEEE Trans. Pattern Anal. Machine Intell., Vol.PAMI–8, pp.699–714, 1986.

[4] A.Rosenfeld, R.A.Hummel and S.W.Zucker, "Scene labeling by relaxation operations," IEEE Trans. Sys., Man, and Cyber., Vol.SMC–6, No.6, pp.420–433, June 1976

[5] S.W.Zucker, R.A.Hummel, and A.Rosenfeld, "An application of relaxation labeling to line and curve enhancement," IEEE Trans. Computers, Vol.C–26, No.4, pp.394–403, April 1977.

[6] A.R.Hanson and E.M.Riseman, "Segmentation of natural scenes," in Computer Vision Systems, New York : Academic Press, pp.129–144, 1978

[7] L.S.Davis and A.Rosenfeld, "Applications of relaxation labelling 2 : Spring–loaded template matching," Technical Report 440, Computer Science Center, Univ. Maryland, 1976.

[8] J.M.Prager, "Extracting and labeling boundary segments in natural scenes," IEEE Trans. Pattern Anal. Machine Intell., Vol.PAMI–2, NO.1, pp.16–27, Jan. 1980.

[9] E.Leung and X.Li, "Parallel processing approaches to edge relaxation," Pattern Recognition, Vol.21, No.6, pp.547–558, 1988.

[10] J.Kittler and J.Foglein, "On compatibility and Support Functions in Probabilistic Relaxation," Computer Vision, Graphics and Image Processing, Vol.34, pp.257–267, 1986.

[11] D.Geman, "Stochastic model for boundary detection," Image and Vision Computing, Vol.5, NO.2, may 1987