Pergamon

Pattern Recognition, Vol. 28, No. 4, pp. 519–536, 1995
Elsevier Science Ltd

0031-3203/95 $9.50 + .00

0031-3203(94)00119-7

AN EFFICIENT EDGE DETECTION ALGORITHM USING RELAXATION LABELING TECHNIQUE*

S. SITHARAMA IYENGAR† and WEIAN DENG
Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803–4020, U.S.A.

(*Received* 15 *September* 1993; *in revised form* 1 *September* 1994; *received for publication* 14 *September* 1994)

Abstract—Edge detection plays an important role in computer vision tasks, and has received considerable attention in image processing literature. To detect edges correctly and precisely, contextual information is needed. How to use contextual information is a key issue. In this paper, we introduce an edge detection method that will use edge contextual information of the whole image efficiency. This new method tries to employ contextual information within a certain distance from the focus pixel at a time. This distance keeps increasing recursively until the edge feature of a pixel is uniquely defined. In this manner, we can minimize the need for contextual information. Experimental results are presented to characterize the performance of our new method in terms of better connectedness of edges and less distortion, and in terms of computational efficiency. A detailed comparison of our method with the context free zero-crossing edge operator that uses optimal exponential filter is discussed in this paper.

Edge detection Relaxation labeling Markov random field Recursive filtering
Probability Configuration dictionary

1. INTRODUCTION

Edge detection plays an important role in computer vision tasks, and has received considerable attention in image processing literature. An edge corresponds to intensity discontinuities in an image. For most machine vision tasks, an edge map is sufficient to conduct further processes such as motion analysis and object recognition. Edges mainly correspond to boundaries of objects of a scene. They may also correspond to images of shadows or surface marks,[1] or the results of noise or blurring. A variety of edge detectors have been proposed. Most of them perform reasonably well for simple noise free images, but tend to fail for noisy images. In our opinion, image smoothing is not the solution. A better way is to make use of edge contextual information.

The ultimate goal of edge detection is to characterize intensity changes of an image in terms of physical process that originate them.[2] It is commonly believed that, to achieve this goal, at least two stages are required: the characterization of intensity changes, and the use of structural and high-level knowledge to find real boundaries.

Intensity changes are detected by differentials of intensity functions. The local maximum of the first order intensity differential and the zero-crossing of the second order intensity differential are the two commonly used characteristics. The results of these differentiation operators are rough edge maps that describe intensity changes of an image. Various techniques have been presented in the literature. Robert's operator[3] and Sobel's operator[4] are examples of these simple edge detection operators.

Canny[5] formulated edge detection problem as an optimization problem. He put forward three objective criteria—good detection, good localization and minimum false alarms— to define an optimal filter. He obtained an optimal one dimensional (1D) operator for step edge detection and found that this optimal operator can be efficiently approximated by the first derivative of Gaussian function. These three criteria were also used by other authors, notably R. Deriche[6] and Shen and Castan,[7,2] to extend the design of optimal filters. The advantage of these two extensions is the recursive feature of the filters. Recursive technique provides an efficient way for image filtering. Both methods use infinite extent filters. Deriche's filters is an infinite extension of Canny's optimal filter and requires five multiplications and five additions for each pixel. Shen and Castan's filter is even more efficient. It is an infinite exponential filter that requires only four multiplications and nine additions for each pixel. As a resemblance to receptive profile of simple cells in mammalian visual systems, Gabor filters have attracted attention recently.[8,9] Gabor filtes are modulation products of Gaussian and sinusoidal signals. Based on Canny's optimal criteria, Mehrotra *et al.*[19] discovered the best performance was a Gabor odd filter and developed an edge detection algorithm based on the filter. Hancock, in his paper,[8] used two filters, a modified Gabor odd filter to detect lines, and a modified Gabor even filter to detect step edges. However, all the

* This work is partly supported by Naval Research Laboratory, under contract N00014-92-J-6003 Remote Sensing Division, Stennis Space Center, MS, U.S.A.
† Author to whom correspondence should be addressed.

above mentioned filters, more or less, have the effect of blurring edges, especially edge junctions. A promising approach to solve this problem is to use nonlinear image filters that encourage intraregion smoothin in preference to inter-region smoothing.[10,11] Perona and Malik[22] proposed anisotropic diffusion for image filtering. The technique is similar to heat flow diffusion phenomenon in physics. Backward flows occur in boundary areas and shapen edges while regions inside boundaries are smoothed. Nitzberg and Shiota[10] further extended this technique. They used regulation to guarantee that diffusion equations had solutions and that corners and T junctures were enhanced.

Although many improvements have been made on image filters (or intensity differentiation operators), using any image filter alone is not sufficient to obtain good edge detection results, especially in noisy situations. One reason is that most filters use models of a single isolated edge. However, the quality of edge detection should not be determined by small differences in smoothing functions. Postprocessing, therefore, is required to further refine rough edge maps obtained from intensity differentiation. One of the techniques to refine rough edge maps is edge tracing.[12,13] Wu, Iyengar and Min[12] investigated edge detection using gradient directional information. In their algorithm, a pixel adjacent to a detected edge pixel, whose magnitude exceeds a given threshold, and whose direction is not perpendicular to that of the edge pixel, is considered as another edge pixel. The algorithm works fine after eliminating many small edges whose length (the number of pixels in an edge) is less than an *ad hoc* threshold value. Ungureanu *et al.*[13] designed another (tracing algorithm that used two bar like control windows. These two windows are perpendicular to each other and are used to walk through edges of an image. They further discussed the VLSI implementation of their algorithm that provided realtime edge refinement. The problem with these approaches is that, they are insensitive to weak edges, and if an edge has a pixel whose magnitude is less than the threshold, they will cut the edge into two smaller edges.

Another technique of edge map refinement is to make use of interaction between edges. Chen and Medioni[14] proposed an edge interaction model to capture interactions between edges within a small neighhorhood area. Initialized with zero-crossings of the signals convolved with a LOG filter, their method iteratively finds new and more accurate edge location by conveying the information from strongly interacting edges. This method yields good results despite the problems of its oversimplified model, its large mask, and its slow convergence rate. Haralick and Lee[15] and Higgins and Hsu[16] also used structural information of neighborhood area to extract edge pixels.

A prospective technique for postprocessing in edge detection is relaxation labeling. Relaxation labeling refers to a family of labeling algorithms, which aim at global interpretations of image objects through iterative update of symbolic label (or meaning) assignments (17). The problem of relaxation labeling was elegantly described by Rosenfeld *et al.*[18] whos proposed four schemes to address it—discrete relaxation, fuzzy relaxation, linear probabilistic relaxation and nonlinear probabilistic relaxation. After that, this area has received much attention. Various approaches have been developed, and they have been successfully applied to many image processing tasks.[19–23] Their ability to convey not only local but also global contextual information from interacting objects makes it a good candidate for edge detection. Kittler and Hancock[21,24,25,8,26] conducted intensive studies on the application of probabilistic relaxation labeling to edge detection. Their approaches employ dictionaries of permissible local edge configurations. A pixel along with its neighborhood is compared with these permissible configurations to estimate the probability that it is assigned a certain label. The goal of their algorithm is to find the globally consistent maximum *a posteriori* probability (MAP) estimate to assign a unique label to each pixel. Noise is modeled as a source of inconsistencies. Interactions among label assignments of pixels are used to eliminate these inconsistencies. However, their method produces good results only in lower signal to noise ratio (SNR) situation. Furthermore, relaxation labeling as a general label assignment framework has a higher time complexity, and takes more time than some other techniques such as the tracing techniques. However, relaxation labeling methods have their advantages. With the ability to link edge segments in local contexts, they produce better edge connectedness. More important, they are easy to be parallelized.

In this paper, we investigate the problem of using relaxation labeling as a post-processing method in edge detection. We propose a new dictionary based relaxation labeling algorithm that has a better noise-suppression performance than Kittler and Hancock's evidence combining formulas. The proposed algorithm uses contextual information as locally as possible. It considers the label context within certain distance from a pixel at a time. This distance keeps increasing until the edge label of a pixel is uniquely determined. We first demonstrate the power of the new method by comparing the results with Kittler and Hancock's algorithm under their assumption that noise is Gaussian distributed. Then, we discuss that initial probability estimate for label assignment is very important to obtain good results for relaxation labeling algorithms, and present a new initial estimation method that is based on histograms of image intensity changes. The advantages of the new method are its robustness to noise, its preservation of corners and T-junctures, and its output edge connectedness.

This paper is organized as follows. The next section includes a description of the new probabilistic relaxation scheme that is derived from Markov Random Field (MRF) theory. An implementation of edge detection using the relaxation method is proposed in Section 3, where the importance of initial probability estimation is discussed, and a new method for initial probability estimation is proposed. The method is based on statistics of a given image. Experimental results that demonstrate the performance of the method are given in Section 4, including comparisons with Kittler and Hancock's relaxation algorithm and Shen and Castan's optimal edge filter. Concluding remarks are given in Section 5.

2. DICTIONARY BASED RELAXATION LABELING

The general idea of probabilistic relaxation labeling is as follows. Suppose a set of objects $V = \{1, 2, \ldots, n\}$ are classified into m categories $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_m\}$, each category j is represented by a label λ_j, $1 \le j \le m$. Suppose further that the category assignments are correlated, and the correlations are described by graph $G = (V, E)$, where an edge in V represents a direct interaction between two objects. Let $P^{(0)}\{x_i = \lambda_j\}$ be an initial estimate of the probability that object i belongs to category λ_j. For each object i, $P^{(0)}\{x_i = \lambda_j\}$ should satisfy $0 \le P^{(0)}\{x_i = \lambda_j\} \le 1$ for $1 \le j \le m$ and $\sum_{j=1}^{m} P^{(0)} \cdot \{x_i = \lambda_j\} = 1$. This initial estimate is calculated from observation vector $\vec{y}_i$ of each object i and $Y = \{\vec{y}_i | i \in V\}$. $\vec{y}_i$ depends on true label assignment x_i and random noise u_i, i.e. $\vec{y}_i = h(x_i, u_i)$, where x_i and u_i are assumed to be independent. The goal of probabilistic relaxation labeling is to find a classification for the objects that is compatible with the initial estimate $P^{(0)}\{x_i = \lambda_j\}$ and the correlation described by graph G. The assignment of a label λ_j to object i is based on *a posteriori* probability $P\{x_i = \lambda_j | Y\}$ of assigning label λ_j to object i under observation set Y.

In relaxation labeling, graph G defines contextual relations among objects. It states that each object interacts with its neighboring objects. The neighborhood of object i is denoted by θi. It is a set of all the adjacent objects of object i. The label of an object depends on the label context of its neighborhood directly. Other objects that are not adjacent provide contextual information in an indirect way. By distinguishing between *directly interacting objects and indirectly interacting objects*, the internal consistency of the transformation function is well maintained.(26)

For example, in the case of edge detection, the object set V consists of all the pixels of an image. Suppose we want to distinguish between edge pixels and non-edge pixels, then, two labels, "edge" and "non-edge" are in the label set Λ. Interactions among objects are described by graph G with an edge set E that connect each pixel with its eight neighboring pixels.

2.1. *The transformation function*

The kernel of probabilistic relaxation labeling is a transformation function, also called projection operator, that describes the relation of a label assignment of an object with the label assignments of its neighboring objects. This function is used to gradually involve more and more contextual information from nearby objects to refine the estimate $P\{x_i = \lambda_j | Y\}$ for label assignment $x_i = \lambda_j$.

To present our transformation function, we first introduce some notations. Associated with each object i is a random variable x_i defined on the set of labels Λ. $X = \{x_1, \ldots, x_n\}$ is the set of random variables for a given problem. $x_i = \lambda_j$ represents the assignment of label λ_j to object i. Set $\omega = \{x_1 = \lambda_{j1}, x_2 = \lambda_{j2}, \ldots, x_n = \lambda_{jn}\}$, called configuration, describes a label assignment for object set V. The set of all configurations is called a configuration space $\Omega = \Lambda^{|X|}$. We use X_i to denote the set of all the random variables associated with objects in V excluding x_i, the random variable for object i, i.e. $X_i = \{x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\}$. Thus, ω_i denotes a configuration for X_i. Ω_i denotes the configuration space of all the ω_is. $\omega_{\partial i}$ denotes a configuration for variable set $X_{\partial i} = \{x_j | j \in \partial i\}$, i.e. neighborhood configuration for object i. $\omega_{\partial i}(l)$ denotes the label assignment for object l in object i's neighborhood under configuration $\omega_{\partial i}$. $\Omega_{\partial i}$ denotes the configuration space of all the configurations $\omega_{\partial i}$ over ∂i.

To estimate the *a posteriori* probability $P\{x_i = \lambda_j | Y\}$, let's consider configuration space Ω_i for object i. For each configuration ω_i in Ω_i, $P\{x_i = \lambda_j, \omega_i | Y\}$ is the probability that a label assignment configuration described by $x_i = \lambda_j$ and ω_i, occurs under observation Y. Since Ω_i contains all the possible label assignment configurations for X_i. $P\{x_i = \lambda_j | Y\}$ can be acquired by adding together all probabilities $P\{x_i = \lambda_j, \omega_i | Y\}$ over Ω_i.

$$P\{x_i = \lambda_j | Y\} = \sum_{\omega_i \in \Omega_i} P\{x_i = \lambda_j, \omega_i | Y\} \tag{1}$$

Using Bayes formula, we get,

$$P\{x_i = \lambda_j | Y\} = \sum_{\omega_i \in \Omega_i} \frac{P\{x_i = \lambda_j | \vec{y}_i\} P\{\omega_i | x = \lambda_j\}}{\sum_{k=1}^{m} P\{x_i = \lambda_k | \vec{y}_i\} P(\omega_i | x_i = \lambda_k\}} \times P\{\omega_i | Y\} \tag{2}$$

In equation 2, ω_i is the configuration of the whole scene. To estimate $P\{x_i = \lambda_j | Y\}$, considering ω as a sample realization of a Markov Random Field X over Ω, then, the locality of MRF

$$P\{x_i = \lambda_j | \omega_i\} = P\{x_i = \lambda_j | \omega_{\partial i}\} \tag{3}$$

can be used to simplify equation 2. Denote the filtered observation after rth iteration as $Y^{(r)} = \{\bar{y}_i^{(r)} | 1 \le i \le n\}$, and the *a posteriori* probability estimate of assigning label λ_j to object i after rth iteration as,

$$P^{(r)}\{x_i = \lambda_j\} = P\{x_i = \lambda_j | Y^{(r)}\} \tag{4}$$

Furthermore, estimate $P\{\omega | Y\}$ by maximum entropy estimation $\Pi_{l \in \partial i} P^{(r)}\{x_l = \omega_{\partial i}(l)\}$. We obtain the following transformation function T,

$$P^{(r+1)}\{x_i = \lambda_j\} = \sum_{\omega_{\partial i} \in \Omega_{\partial i}} W_{\omega_{\partial i}}^{(r)} E^{(r)}(i, j, \omega_{\partial i}) \tag{5}$$

Where

$$E^{(r)}(i, j, \omega_{\partial i}) = \frac{P^{(r)}\{x_i = \lambda_j\} P\{\omega_{\partial i} | x_i = \lambda_j\}}{\sum_{k=1}^{m} P^{(r)}\{x_i = \lambda_k\} P\{\omega_{\partial i} | x_i = \lambda_k\}} \tag{6}$$

$$W^{(r)}_{\omega\partial i} = \prod_{l \in \partial i} P^{(r)}\{x_l = \omega_{\partial i}(l)\} \quad (7)$$

For detailed derivation of this transformation function. Please refer to,[27] Function T provides a new approach for probabilistic relaxation. A new estimate for $P\{x_i = \lambda_j | Y\}$ is calculated by adding together all the supports from object i's neighborhood configurations. Since the denominator of $E(i, j, \omega_{\partial i})$ is only a normalization factor, the support from a single configuration is proportional to the *a priori* probability of the configuration and the probability estimates of the label assignments for objects in current configuration. Though applying transformation function T once conveys only the constraints of inter-object relations among neighboring objects, using T recursively can convey contextual information from indirect interacting objects, and the estimation of $P\{x_i = \lambda_i | Y\}$ will become more and more precise. Because of the use of Maximum Entropy estimate for $P\{\omega_{\partial i} | Y\}$, the transformation function is proved to converge to local optimal according to different initial estimate.[27]

This new relaxation scheme satisfies the two probabilities axioms $0 \le P^{(r)}\{x_i = \lambda_j\} \le 1$, for all r, i and j, and $\sum_{j=1}^{m} P^{(r)}\{x_i = \lambda_j\} = 1$, for all r and i. These properties guarantee that T is a transformation from mn dimensional probability space to itself. By the Bronwer fixed point theorem, the relaxation rule converges.

The process to find a label assignment for object set V has two steps. First, the conditional probability $P\{x_i = \lambda_j | Y\}$ is estimated. Then, a label is assigned to object i by maximizing *a posteriori* probability (MAP), i.e. Assign λ_j to object i if

$$P\{x_i = \lambda_j | Y\} = \max_{k=1}^{m} P\{x_k = \lambda_{jk} | Y\} \quad (8)$$

The estimation for $P\{x_i = \lambda_j | Y\}$ begins with the information carried by object i only, i.e. the noisy observation $\vec{y}_i$. $P\{x_i = \lambda_j | \vec{y}_i\}$ is used to calculate an initial estimate for $P\{x_i = \lambda_j | Y\}$,

$$P\{x_i = \lambda_j | \vec{y}_i\} = \frac{P\{\vec{y}_i | x_i = \lambda_j\} P\{x_i = \lambda_j\}}{P\{\vec{y}_i\}} \quad (9)$$

$$P\{\vec{y}_i\} = \sum_{j=1}^{m} P\{\vec{y}_i | x_i = \lambda_j\} P\{x_i = \lambda_j\} \quad (10)$$

Since $\vec{y}_i$ contains noise, this estimation may cause labeling errors. However, these labeling errors can be detected because they are inconsistent within the context of label assignment of other objects. Therefore, transformation function T is recursively used to refine the estimation of $P\{x_i = \lambda_j | Y\}$.

2.2. *The dictionary model*

Based on transformation function T, the outline of the algorithm is as follows: First, we initialize vector $P^{(0)}$ in the mn dimensional probability vector space without considering contextual relations, and use maximum *a posteriori* probability principle to assign an initial label to each object. Then, transformation T is recursively applied to vector $P^{(r)}$ to obtain a refined estimate $P^{(r+1)}$ that involves more contextual information. Each time, a new label assignment is found by maximizing *a posteriori* probabilities. Termination of the algorithm is guaranteed by the convergence of the transformation function discussed in Section 2.1. Once label assignments are stable, the algorithm stops.

However, the complexity to calculate transformation T is proportional to $|\Omega_{\partial i}|$, the number of configurations of object i's neighborhood. In the worst case, a neighborhood may consists of all other objects in the system, then the number of neighborhood configurations $|\Omega_{\partial i}| = m^n$ and is exponent in the number of objects. This time complexity can be cut down however. First, the size of neighborhoods is usually much smaller than the size of the whole object set, e.g., in the case of image processing, neighborhood size is usually 3×3 or 5×5 etc. However, even with 3×3 neighborhood, the number of all configurations is still m^9, where m is the number of labels in the label set. However, most applications are well-structured. Thus, most of the configurations are physically impossible. A small set of permissible configurations results. *Permissible configurations* are the configurations which can occur in the given application in an ideal situation. As an example, in edge detection application with a neighborhood of 3×3 lattice and a set of five labels (Four labels for the four different edge directions and a non-edge label), the number of all configurations is $5^9 \approx 2 \times 10^6$. But only 165 of them are permissible configurations. This suggests that, for an application, with the construction of configuration dictionary that excludes all impossible configurations, relaxation labeling method can be implemented efficiently.

Transformation function T contains two types of probabilities. One of them has the form $P^{(r)}\{x_i = \lambda_j\}$ that is the current estimate of a label assignment. The other type has the form $P\{\omega_{\partial i} | x_i = \lambda_j\}$ and is *a priori* probability that indicates how often configuration $\omega_{\partial i}$ occurs when object i is assigned a particular label λ_j. Transformation function T is the sum of supports from all configurations of an object's neighborhood. This can be further divided into two sums.

$$\begin{aligned} P^{(s+1)}\{x_i = \lambda_j\} &= \sum_{\omega_{\partial i} \in \Omega_{\partial i}} W^{(r)}_{\omega_{\partial i}} E^{(r)}(i, j, \omega_{\partial i}) \\ &= \sum_{\omega_{\partial i} \in \Omega_{\partial i}} W^{(r)}_{\omega_{\partial i}} E^{(r)}(i, j, \omega_{\partial i}) \\ &\quad + \sum_{\omega_{\partial i} \in \Omega^{D}_{\partial i}} W^{(r)}_{\omega_{\partial i}} E^{(r)}(i, j, \omega_{\partial i}) \end{aligned} \quad (11)$$

where $\Omega^{D}_{\partial i}$, the "don't care" configuration set,[21] contains configurations such that $P\{\omega_{\partial i} | x_i = \lambda_j\} = P\{\omega_{\partial i}\}$ for all label assignment λ_j, i.e. under the neighborhood configuration $\omega_{\partial i}$, $P\{x_i = \lambda_j\}$ is independent from its neighbors. $\Omega^{C}_{\partial i}$, the "care" configuration set, is the complement of $\Omega^{D}_{\partial i}, \Omega_{\partial i} = \Omega^{C}_{\partial i} + \Omega^{D}_{\partial i}$. Configurations in $\Omega^{C}_{\partial i}$ are in facour of changing the probability $P\{x_i = \lambda_j\}$. Equation 11 provides an approach to implement the relaxation scheme. However, this implementation has inherent difficulties. Because

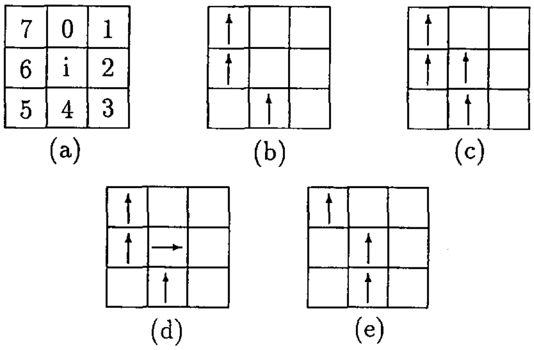


Fig. 1. Possible supporting configurations.

the number of configurations $|\Omega_{\partial i}|$ is huge. For example, a 3×3 neighborhood with five labels has $|\Omega_{\partial i}| = 5^9 \approx 3 \times 10^6$ configurations. If $|\Omega_{\partial i}| \approx |\Omega^C_{\partial i}|$, the computation is very expensive since there are large number of configurations in the calculation. Usually we prefer to have $|\Omega_{\partial i}| \gg |\Omega^C_{\partial i}|$ i.e., we choose only a few "care" configurations that have significant contributions to label assignments. Then, the expense to calculate T is acceptable. However, the rate of convergence will be very slow because $|\Omega^D_{\partial i}| \gg |\Omega^C_{\partial i}|$, i.e. the factor for change $\sum_{\omega_{\partial i} \in \Omega^C_{\partial i}}$ is much smaller than the factor for stable $\sum_{\omega_{\partial i} \in \Omega^D_{\partial i}}$.

Another method has been proposed[8] that considers $P\{x_i = \lambda_j, \omega_{\partial i}\}$ as zero for configuration $\omega_{\partial i}$ that has little contribution to label refinement, for example, those in $\Omega^D_{\partial i}$. In this way, we get a small set of configurations that have significant influences to label assignments. We call it the Influential Configuration Set (ICS) $\Omega^I_{\partial i}$. Then, we have

$$P^{(r+1)}\{x_i = \lambda_j\} = \sum_{\omega_{\partial i} \in \Omega^I_{\partial i}} W^{(r)}_{\omega_{\partial i}} E^{(r)}(i, j, \omega_{\partial i}) \quad (12)$$

For the Influential Configuration Set $\Omega^I_{\partial i}$, dictionary method can be employed to provide an efficient search. Dictionary method was successfully used by Kittler and Hancock. Here, we use a new construction method that includes more configurations. Each object i has its own dictionary D_i. D_i consists of configurations that come from ICS $\Omega^I_{\partial i}$. Dictionary D_i is a table with s rows and m columns, where s is the number of neighborhood settings and m is the number of possible labels. $D_i(\lambda_j)$ denotes the column corresponding to the assignment of λ_j to object i. Let λ^k_{jl} be the label on object $l, l = i$ of the kth neighborhood configuration item in column $D_i(\lambda_j)$. The kth configuration in $D_i(\lambda_j)$ is

$$C^k_i(\lambda_i) = \{x_l = \lambda^k_{jl}, \quad l \in \partial i | x_i = \lambda_j\} \quad (13)$$

And the probability associated with $C^k_i(\lambda_j)$ is $P\{x_l = \lambda^k_l, \; l \in \partial i | x_i = \lambda_j\}$, the conditional probability that a configuration of i's neighborhood ($x_l = \lambda^k_l, l \in \partial i$) occurs when $x_i = \lambda_j$.

Our method differ from Kittler from Hancock's[26] in that, we consider the ICS $\Omega^I_{\partial i}$. It includes not only all the permissible configurations that are used in K & H's method, but also other configurations that have significant influence over label assignments. For example, in image edge detection, for a pixel i, we take the eight surrounding pixels as its neighboring pixels (See Fig. 1a). All the possible one pixel edge patterns across this 3×3 area are permissible configurations. One permissible configuration in this setting is showed in Fig. 1b. Here, arrows indicate directions of edge pixels and blank spaces non-edge pixels. The label set contains five labels: four different edge directions $\rightarrow$, $\uparrow$, $\leftarrow$ and $\downarrow$, and a non-edge label ε. This permissible configuration $\{x_0 = \varepsilon, \; x_1 = \varepsilon, \; x_2 = \varepsilon, \; x_3 = \varepsilon, \; x_4 = \uparrow, \; x_5 = \varepsilon, \; x_6 = \uparrow, \; x_7 = \uparrow, \; x_i = \varepsilon\}$ is an entry in the dictionary and has a conditional probability $P\{x_0 = \varepsilon, \; x_1 = \varepsilon, \; x_2 = \varepsilon, \; x_3 = \varepsilon, \; x_4 = \uparrow, \; x_5 = \varepsilon, \; x_6 = \uparrow, \; x_7 = \uparrow, \; x_i = \varepsilon\}$ associated with it.

Now, let's assign other labels for x_i in the same neighborhood setting. For example, $x_i = \uparrow$ in Fig. 1c and $x_i = \rightarrow$ in Fig. 1d. In Fig. 1c, it is possible that in the previous relaxation iterations, there is a label error for pixel 6, i.e. its label should be ε not $\uparrow$. Because this error may be corrected later in the relaxation process, the configuration provides a support to assign $\uparrow$ to pixel i. However, in Fig. 1d, the neighborhood configuration surely has no support for $x_i = \rightarrow$ because no premissible configurations with $x_i = \rightarrow$ can be obtained by removing some of pixel i's neighboring edge pixels. Thus, we further divide the category of physically impossible configurations like those in Fig. 1c and d into two categories: those that are included in ICS because they have contributions to certain label assignments for pixels, i.e. Fig. 1c. We call them *possible supporting configurations.* The other category contains configurations such as the one in Fig. 1d.

Therefore, in our scheme, the set of configurations that we are used i.e., ICS, consists of:

(1) *permissible configurations* which occur in the ideally situation, and;

(2) the configurations which would lead to some permissible configurations if some of its neighboring objects' labels changes while the label for the object i remains the same. We call this kind of configurations as *possible supporting configurations.*

Based on the proposed transformation function T and the dictionary model, the relaxation labeling algorithm is depicted in Fig. 2. First, an initial estimate $P^{(0)}\{x_i = \lambda_j\}$ for $P\{x_i = \lambda_j | Y\}$ is calculated from observation vector $\vec{y}_i$ for object i by equations 9 and 10. Based on $P^{(0)}\{x_i = \lambda_j\}$, an initial label assignment is selected by maximizing *a posteriori* probability (equation 8). The result is a rough label assignment for object system V. The relaxation carrying out by equation 12 where the influential configurations and their associated conditional probabilities are obtained by looking up dictionary. $D_i(\lambda_j)$ for object i and its assigned label λ_j. In each relaxation iteration, label assignments are updated by MAP rule (equation 8). This procedure ends with refined label assignments.

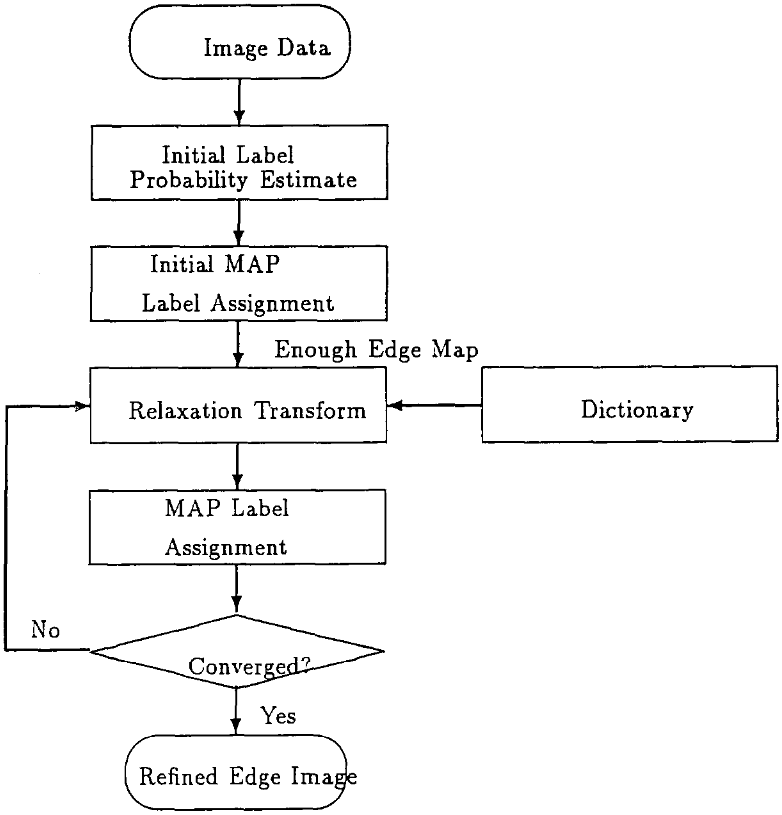


Fig. 2. The relaxation algorithm.

3. RELAXATION BASED EDGE DETECTION

In traditional edge detection, differentials are used. However, differentiations are very sensitive to noise. Although this problem can be eased by smoothing, smoothing can also eliminate edge features and degrade resolution capabilities of edge detectors simultaneously.

Using relaxation labeling as a postprocessing step is a prospective solution. First, a traditional differential operator is employed to obtain an initial edge assignment for every pixel. This edge detector should preserve as many edges as possible. A dictionary of configurations in 3×3 neighborhood of each pixel is then constructed and used in probabilistic relaxation labeling algorithm to correct the erroneously labeled pixels. The effect of this postprocessing is to remove noise and to acquire the refined one pixel wide edge map of the given image.

To develop an edge detection algorithm from the dictionary based relaxation scheme, two problems need to be addressed:

(1) how to calculate initial label probability estimates;

(2) how to find the configurations for the dictionary and to calculate the *a priori* conditional probabilities associated with them.

3.1. *Previous results*

In our previous preliminary investigation,[4] we described a relaxation labeling scheme under the assumption that noise was Gaussian distributed. The smallest differential operators (1×2 and 2×1 windows) were used. In that implementation, each pixel (u, v) had an observation vector $\vec{y}_{(u,v)}$ with two first order partial differentials c_u and c_v of the observed intensity function $g'(u, v)$,

$$c_u = g'(u+1, v) - g'(u, v)$$
$$c_v = g'(u, v+1) - g'(u, v)$$

The Gaussian noise was assumed to have a zero mean and a standard deviation of σ. For non-edge pixel (u, v), *pixels*$(u+1, v)$, $(u, v+1)$ and (u, v) belong to the same image segment and should have the same standard deviation σ. Thus, *a priori* probability $P\{c_u, c_v | x = \varepsilon\}$ was calculated by,

$$P\{c_u, c_v | x = \varepsilon\} = \frac{1}{2\sqrt{3}\pi\sigma^2} \exp\left[\frac{c_u^2 + c_v^2 - c_u c_v}{3\sigma^2}\right] \quad (14)$$

The initial label assignment probabilities were then computed by first estimating $P\{x = \varepsilon | c_u, c_v\}$ from equation 14 and Bayes formula, and then distributing the residual among the four edge labels (upward, downward, rightward and leftward).

Five labels were used to classify pixels. They were ε for non-edge pixel, $\uparrow$ for upward edge pixel, $\rightarrow$ for rightward edge pixel, $\downarrow$ for downward edge pixel, and $\leftarrow$ for leftward edge pixel. The criteria to find the permissible configurations were:

(1) edges are all closed;
(2) edges are all one pixel wide;
(3) edges are all continuous.

One hundred and sixty five permissible configurations were found based on these criteria. Of these 97 had label ε for the center pixel. And for each of the four edge labels assigned to center pixel, there were 17 permissible configurations. All permissible configurations were considered equally likely thus, for each permissible configuration ω,

$$P\{x_i = \lambda_j, \omega\} = \frac{1}{165}.$$

Two kinds of images were used (an artifact image with additional Gaussian noise of different level, and some natural images) to compare the algorithm with K & H's.[8] The results showed that both methods are very good in preserving corner and edge connectivity, but our method has a better noise suppression capability. See[4] for details. However, a primary goal to obtain single pixel edge was not fully accomplished. For example, Fig. 11b and d are the edge outputs obtained from Kittler and Hancock's method and our relaxation method. In both cases, the edges around the fluorescent lamps are not very well constructed.

3.2. *The problem of initial estimation*

Although designing an fast convergent update function is a major step in developing probabilistic relaxation algorithm, the problem of initial assignment is also crucial. It is true that applying contextual information efficiently through the update procedure can eliminate ambiguities from imprecise initial label assignments. However, if initial assignments contain too many label errors, label contextual information may not be enough to eliminate all of them. Indeed that was the problem of our previous initial label assignment estimation approach. More precisely, the problem stems from:

(1) the assumption that noise is Gaussian distributed may not work in real world applications;

(2) it only uses first order differences as the observation vector.

Based on these observations, we propose a new approach to compute initial label assignments. First, histogram $h(l)$ of first order difference of an image is used instead of Gaussian distribution assumption. l Denotes the absolute change in gray level. Secondly, second order difference and first order difference are incorporated to obtain a better initial guess of label assignment.

For a particular image, its noise may not be Gaussian distributed. A histogram, on the other hand, is a statistic of a given image and better reflects the intensity distribution of the image. Thus, a histogram provides a better estimation. In our method, the histogram of first order difference of intensity level is used because the probability estimation of initial edge label assignment is based on intensity changes.

Zero-crossings of second order difference of an image has been proved to be a good estimate of edge points. If a pixel is not a zero-crossing point, this pixel is not an edge pixel. However, a zero-crossing point may not be an edge pixel. The degree of intensity changes and label contextual information are needed to further refine the edge map that was derived from zero-crossings.

To get zero-crossings of a given image, we adopt J. Shen and S. Castan's Exponential recursive filtering approach.[5,26,27] Their exponential filter (Fig. 3) has infinitely large window size and can be realized by a

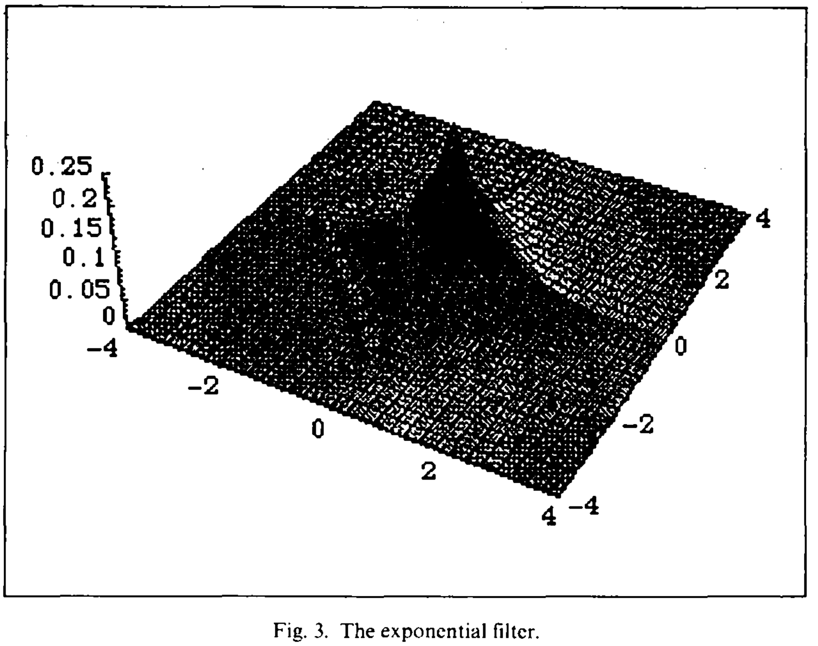


Fig. 3. The exponential filter.

simple and efficient recursive algorithm. An excellent feature of this filter is that the limited Laplacian of an input image filtered by this filter can be computed from the difference between the input and the output of the filter. Thus, second order difference of an image can be calculated efficiently. The 1D exponential filter has the form

$$f(x) = \left[\frac{a_0^2}{1-(1-a_0)^2}\right](1-a_0)^{|x|}$$

and can be implemented by two recurrent relation: first, scan from left to right using

$$y_1(i) = y_1(i-1) + a_0[x(i) - y_1(i-1)]$$

Then, scan from right to left using

$$y_2(i) + y_2(i-1) + a_0[y_1(i) - y_2(i-1)]$$

Since the filter is decomposable, a 2D exponential filter can be implemented by two 1D filters: one in x direction and the other in y direction. And the second order difference is calculated from the difference between filter output and filter input.

Binary Laplacian Image (BLI) is employed to find zero-crossings. A BLI is a binary image where a pixel gets value 1 (0) if the corresponding second order difference is non-negative (negative). The pixels lay in boundaries of 1-segments are zero-crossing pixels. Since very small, isolated 1(10)-segments in BLI are the results of random noise, an additional step is used to eliminate all small isolated segments, like those that have less than five pixels.

To get a better estimate of initial label assignment, a histogram $h(l)$ of the absolute intensity level changes of the image is calculated. This histogram is then used to estimated probabilities $P\{x_i = \text{"edge"}\}$ for initial assignments for the "edge" labels,

$$P\{x_i = \text{"edge"}\} = \begin{cases} 0 & x_i \text{ is not a boundary pixel of 1-segment} \\ 0 & \text{if } c_i \le 6 \\ h(c) & \text{otherwise} \end{cases}$$

where $c_i = \max(|c_{x_i}|, |c_{y_i}|)$, and the estimated probabilities for initial assignments of the "non-edge" label are

$$P\{x_i = \text{"non-edge"}\} = 1 - P\{x_i = \text{"edge"}\}$$

If pixel i is not a zero crossing point, the probability that i's label is "edge" is zero. Otherwise, the intensity difference along c axis c_{x_i}, and the intensity difference along y axis c_{y_i}, is calculated. The maximum value c_i between the absolute value of c_{x_i} and that of c_{y_i} is used as the measurement of an intensity change for the pixel. We use maximum value c_i in the calculation instead of averaging $|c_{x_i}|$ and $|c_{y_i}|$, because a significant intensity change in any direction is sufficient to consider the pixel as an edge pixel. It has been found that intensity changes below six gray levels [the *just noticeable difference* (JND)] in 256 gray level scaled images are not detectable by human eyes.[32] Therefore, in the case that c_i is less than or equal to gray level 6, the pixel is considered as a non-edge pixel.

To summarize, the procedure of label assignment initialization is shown in Fig. 4. It first uses the exponential filter to calculate second order intensity difference of an image. With the help of the BLI, all zero crossing points are located. A histogram of the first

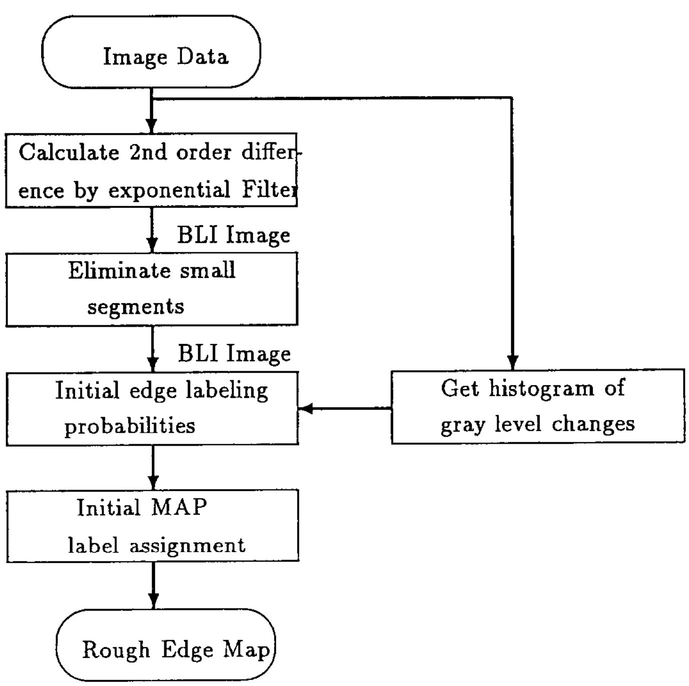


Fig. 4. The initialization procedure.

order intensity difference of the image is then calculated and is used to assign initial probability estimates for initial label assignments. An initial label is assigned to pixel i by:

$$x_i = \begin{cases} \text{“edge”} & \text{If } P\{x_i = \text{“edge”}\} \geq \\ & \quad P\{x_i = \text{“non-edge”}\} \\ \text{“non-edge”} & \text{Otherwise} \end{cases}$$

3.3. *The edge dictionary*

To complete the edge detection algorithm, a dictionary of configurations needs to be constructed. Here, only two labels, “edge” and “non-edge” are used to classify pixels. No direction information is employed in the current implementation of the algorithm. The permissible configurations in this application are obtained from ideal edge maps where the three criteria listed in Section 3.1 are satisfied. Nine basic permissible configurations are found which are listed in Fig. 5 (a dot in a pixel indicates an edge pixel). By rotation, reversal and reflection, we get 41 permissible configurations, out of which, 12 configurations support the assignment of “edge” label and 29 configurations support “non-edge” label. In this study, all permissible configurations are considered equally likely. Thus each permissible configuration $\omega - \{x_i = \lambda_j, x_l = \lambda_l, l \in \partial i\}$ is associated with a probability of

$$P\{\omega\} = \frac{1}{41}.$$

The probability associated with dictionary item $C_i^k(\lambda_j) = \{x_l = \lambda_l^k,\ l \in \partial i | x_i = \lambda_j\}$ is $P\{x_l = \lambda_l^k,\ l \in \partial i | x_i = \lambda_j\}$, the conditional probability that configuration $\{x_i = \lambda_j, x_l = \lambda_l^k,\ l \in \partial i\}$ occurs when $x_i = \lambda_j$. This probability is calculated as follows. Let Ω^p be the set of all permissible configurations. First, we obtain the *a priori* probability $P\{x_i = \lambda_j\}$ by adding together the probabilities

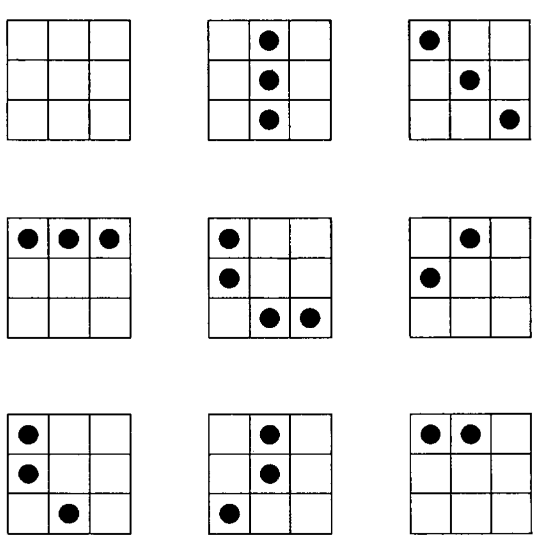


Fig. 5. Permissible configurations for edge detection.

of all permissible configurations where $x_i = \lambda_j$.

$$P\{x_i = \lambda_j\} = \sum_{\omega \in \Omega^p} \tag{15}$$

where $x_i = \lambda_j$

Then, the probability for each permissible configuration item in $D_i(\lambda_j)$ is then calculated by

$$P\{C_i^k(\lambda_j)\} = \frac{P\{x_i = \lambda_j, x_l = \lambda_l^k, l \in \partial i\}}{P\{x_i = \lambda_j\}} \tag{16}$$

Each of these 41 configurations has a distinct neighborhood setting. Thus, there are 41 neighborhood settings and the dictionary has 82 configurations, 41 permissible configurations and 41 possible supporting configurations.

Given all the permissible configurations and their probabilities $P\{x_i = \lambda_j, x_l = \lambda_l^k, l \in \partial i\}$, we need to compute probabilities for possible supporting configurations. In the relaxation process, a possible supporting configuration occurs when labeling errors are present. Thus, it is natural to consider possible supporting configurations as corrupted permissible configurations. In,[10] Kittler and Hancock proposed a label error process for discrete relaxation. They derived formulas to estimate the probability for any possible configuration from permissible configurations in their attempt to develop a discrete relaxation algorithm. The idea in this estimation is to add together the likelihood of the current label configuration with all the permissible configurations.

Assume that label errors occur with equal probability p_e. The likelihood of a possible supporting label configuration $\omega = \{x_i = \lambda_j, x_l = \lambda_l^k, l \in \partial i\}$ with a permissible configuration $\omega^p \in \Omega^p$ is described by

$$P\{\omega | \omega^p\} = (1 - p_e)^{|\partial i| - K(\omega, \omega^p)} p_e^{K(\omega, \omega^p)} \tag{17}$$

$|\partial i|$ is the number of neighborhood objects, $K(\omega, \omega^p)$ is the number of labels that are different between a possible supporting configuration ω and a given permissible configuration ω^p. This likelihood is called *neighborhood transition probability*. The probability for a possible supporting configuration is the summation of all neighborhood transition probabilities over all permissible configurations.

$$P\{x_i = \lambda_j, \omega_{\partial i}\} = \sum_{\omega \in \Omega^p} P\{x_i = \lambda_j, \omega_{\partial i} | \omega\} P\{x_i = \lambda_j, \omega_{\partial i}\} \tag{18}$$

where $x_i = \lambda_j$.

4. EXPERIMENTS

To understand the performance of the new algorithm, we examine the behavior of the algorithm on natural images as well as artifact images. These experiments are intended to test the robustness of the new dictionary based probabilistic relaxation labeling algorithm. We compared our results with Kittler and Hancock’s probabilistic relaxation algorithm (called K & H) and Shen and Castan’s optimal exponential edge detector, the

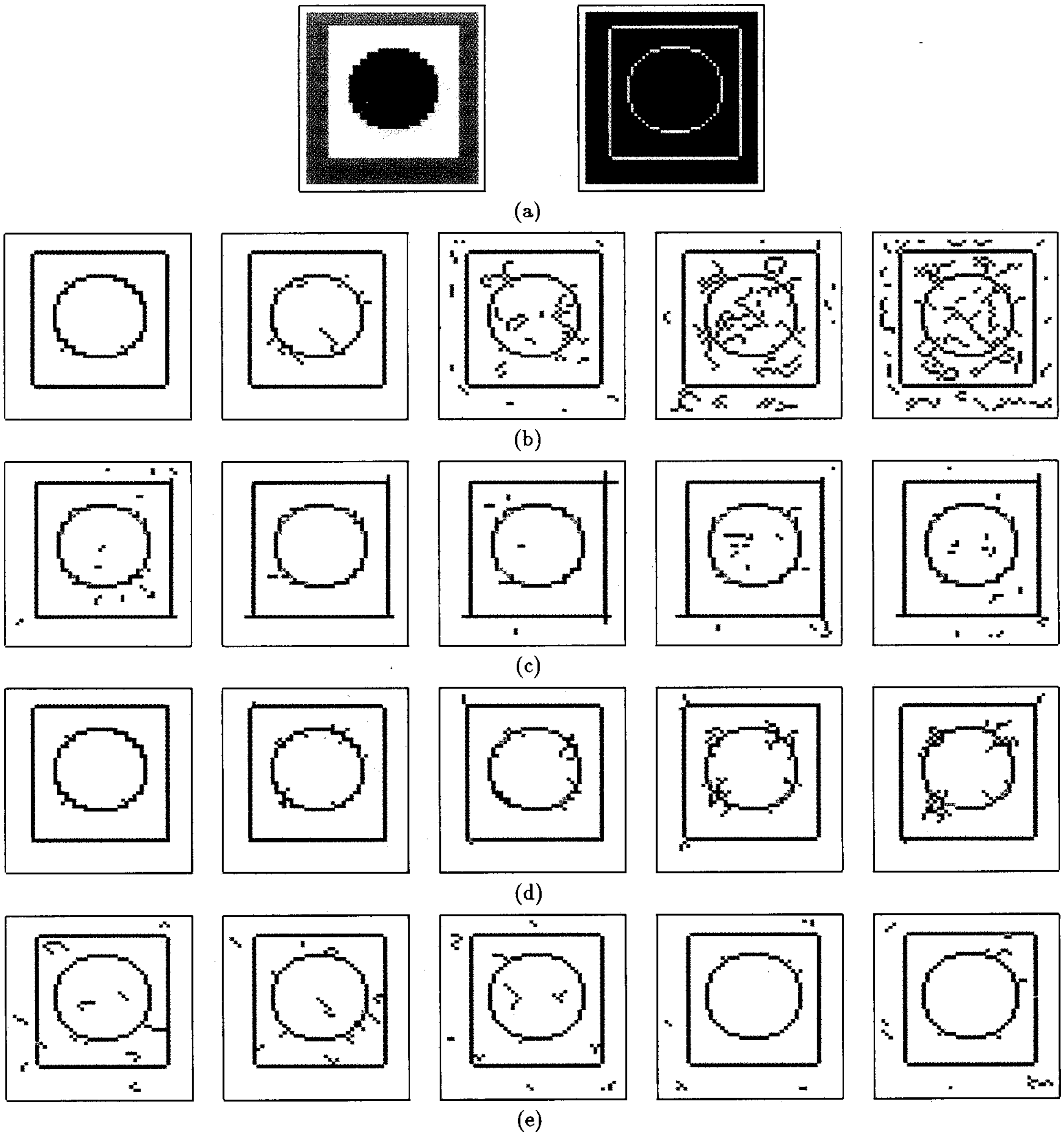


Fig. 6. Synthetic image (0–255) with additional Gaussian noise ($\sigma = 20$, 40, 60, 80, 100, 120, 140 and 180. (a) Is the original image without noise; (b) is the result of Kittler and Hancock's algorithm; (c) is the result of Shen and Castan's algorithm; (d) is the result of our first algorithm, and (e) is the result of our new algorithm.

edge detection algorithm SDEF in image processing system Khoros. We developed two versions of our relaxation algorithm to examine the importance of initial label assignment estimation: one of them uses the initialization method proposed by Kittler and Hancock (Called Algol); the other follows the method described in last section (Called Algo2). All the algorithms are programmed on SiliconGraphics in C.

In the following presentation, for algorithms K & H, Algo1, and Algo2, all the edge outputs are collected after 10 iterations. The figures show the final maximum *a posteriori* label assignments. Black pixels correspond to non-edge pixels and white pixels are edge pixels. It should be noticed that all the results shown are obtained from initial probability assignments after applying the relaxation transformation functions 10 times. No post-processing like linking, thinning, or cleaning, etc. is done.

A well structured simulated image (50×50 in dimension with a square and a circle) is used. Within the circle, the gray level is 56. Outside the circle but inside the square, the gray level is 231. Outside the square, the gray level is 115. This perfect image is then mixed with independent Gaussian noises using Khoros. These noises have a mean of zero and S.D.'s of 20, 60, 100, 140 and 180, respectively. The artifact image and its standard one pixel-wide eight-connected edge map are shown in Fig. 6a. This is to test the performance

Table 1. Number of mislabeled pixels

σ	20	60	100	140	180
K & H	22	30	100	198	162
SDEF	65	48	61	77	68
Algo1 (proposed)	25	31	46	77	81
Algo2 (proposed)	49	37	41	14	32

Table 2. Number of break points

σ	20	60	100	140	180
K & H	0	1	0	0	0
SDEF	4	5	3	6	3
Algo1 (proposed)	0	0	1	0	1
Algo2 (proposed)	0	0	0	0	1

of the algorithms under different noisy situations and its ability to detect edges of various orientations and edges with high curvature. Figure 6 shows the result of these algorithms. In these figures, a black pixel indicates a correct labeling of an edge pixel. A red pixel is a pixel mislabeled as an edge pixel. A light blue pixel is a pixel mislabeled as a non-edge pixel.

For noise suppression both SDEF and Algo2 have good noise resistance, and work consistently under different noise level. The performances of K & H and Algo1 are affected by noise. Noise with higher standard deviation causes more labeling errors. K & H method obtains more error labeled edge-pixels. Table 1 shows the number of mislabeled pixels for all the output images.

For edge connectedness, if error labeled edge pixels are ignored, K & H and Algo2 capture the contours of the standard edge map (Fig. 6a) quite precisely and the results are almost free of distortions. Algo1 obtains the standard contour without distortion when $\sigma = 20$ and $\sigma = 60$. The edge outputs are distorted for $\sigma \geq 100$ and break points are also presented. SDEF has break points in al the cases and the contour for the circle tend to deviates from the standard edge map. Figure 7 shows that the enlarged edge maps for $\sigma = 100$ clearly demonstrates the correctness of the edge outputs. Table 2 summarizes the number of breaks in each case.

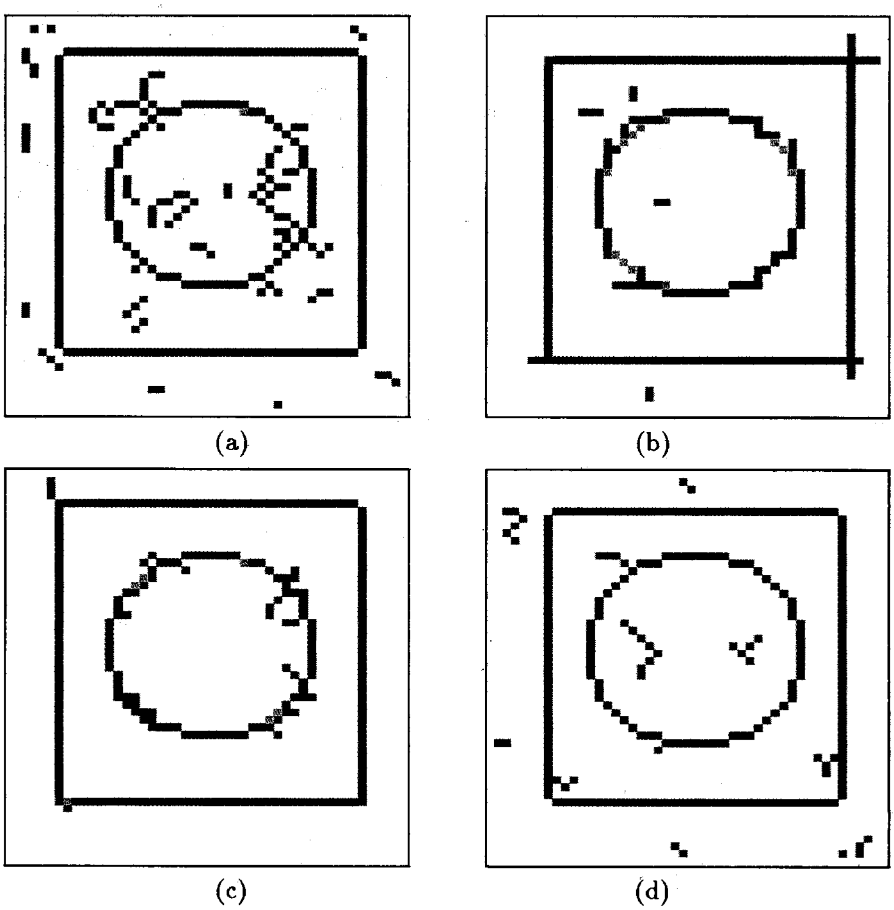


Fig. 7. Synthetic image (0–255) with additional Gaussian noise $\sigma = 100$. (a) Is the result of Kittler and Hancock's algorithm; (b) is the result of Shen and Castan's algorithm; (c) is the result of our first algorithm and (d) is the result of our new algorithm.

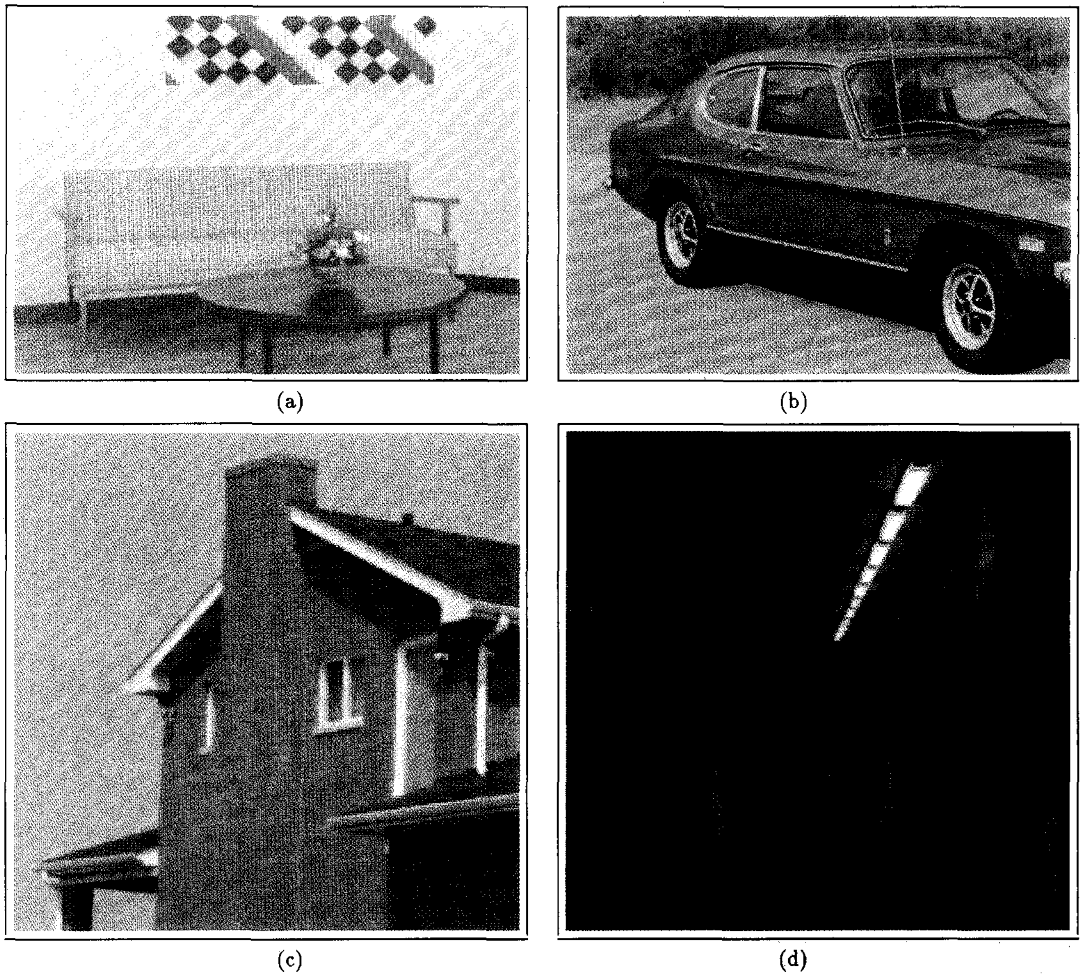


Figure 8: Four natural images: (a) a corner of an office; (b) a car; (c) a house; (d) an indoor scene.

The differences in both noise resistance and contour perfectness between the outputs from Algo1 (Fig. 6d) and those from Algo2 (Fig. 6e) reveal the importance of label assignment initialization.

To assess the effectiveness of our method to correctly label edges for natural images, four images from an image base in the University of Massachusetts are used (Fig. 8).

Figure 9 is the edge maps for the office scene (Fig. 8a). For the simple patterns on the wall. SDEF and Algo2 obtain clear one pixel wide edges. However, the outputs from K & H and Algo1 are not one pixel wide. An example of weak contrast edges is the seat of the couch. Figure 10 highlights this portion of the output. The outputs from K & H and Algo2 capture more weak contrast edges.

The results of processing Fig. 8c are shown in Fig. 11 K & H's method acquires more edges and also retains more noisy pixels. SDEF and Algo2 obtain clearer edge maps, especially in the areas near the fluorescent lamps. However, these two methods fail to capture the juncture between the left wall and the ceiling. SDEF also fails to capture the junctures between the wall and the floor.

Figure 8b and d are two more examples of natural images, where all methods perform reasonable well. In both cases, K & H method retains more noise. K & H and Algo1 cannot obtain one pixel wide edges in some areas, the wheels of the car and the eaves of the house. These two images also reveal that relaxation methods, through the use of neighborhood label context, achieve better edge connectedness. This is demonstrated by the stripes on the body of the car and the eaves of the lower roofs of the house, where the relaxation methods obtain connected lines while SDEF gets dashes.

To summarize, these experiments show that:

(1) SDEF and Algo2 have better noise resistance;

(2) relaxation methods, K & H, Algo1 and Algo2, obtain better edge connectedness and better contour;

(3) K & H and Algo2 achieve better weak edge detection;

(4) The estimation of initial label assignments is

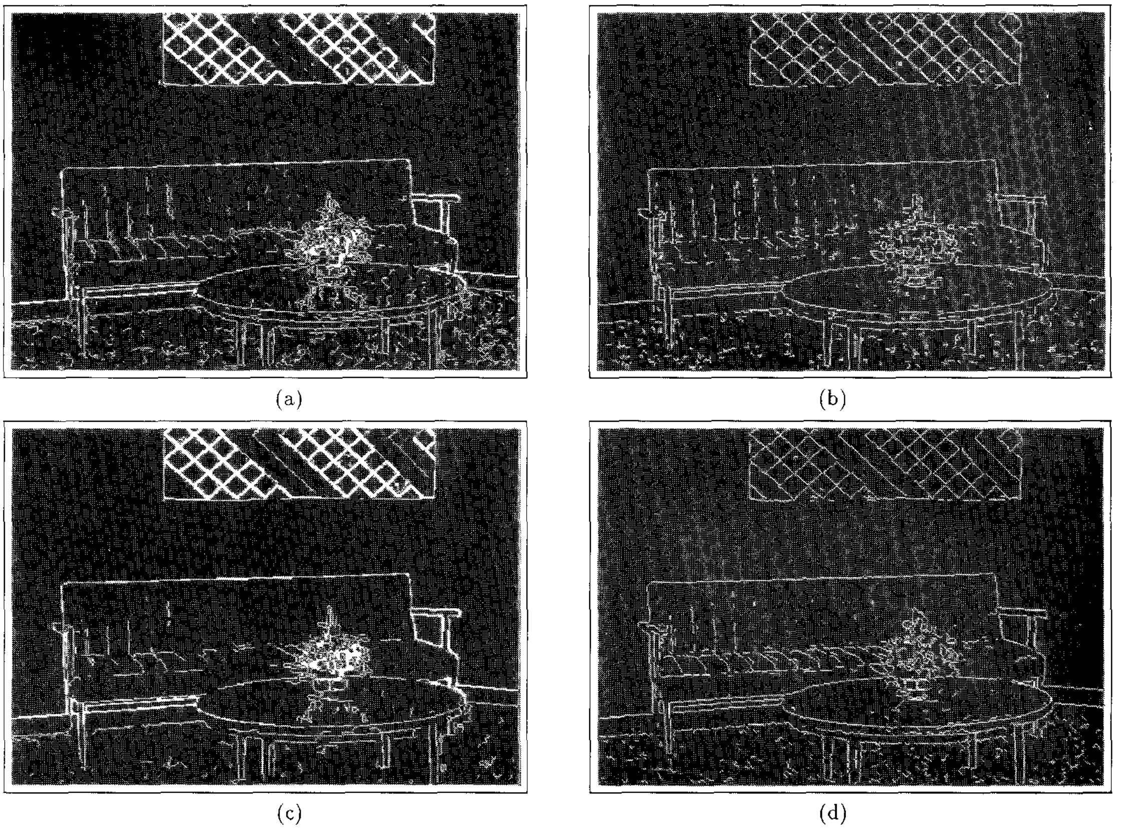
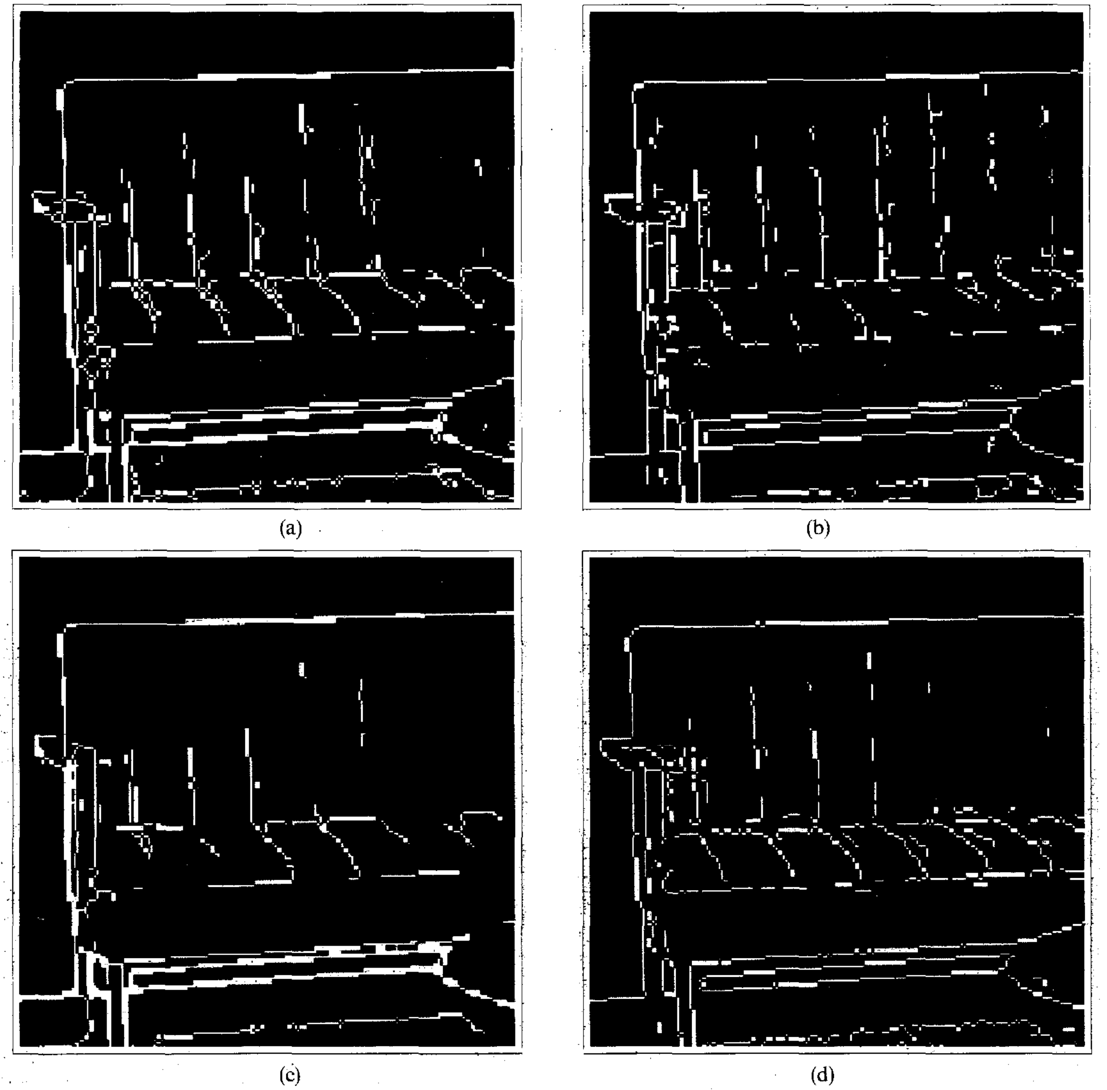


Fig. 9. The scene of an office. (a) Is from Kittler and Hancock's algorithm: (b) is from Shen and Castan's algorithm: (c) is from our algorithm Algo1 and (d) is from our algorithm Algo2.



(a) (b)

(c) (d)

Fig. 10. Weak contrast edges. (a) Is from Kittler and Hancock's algorithm; (b) is from Shcn and Castan's algorithm: (c) is from our algorithm Algo1 and (d) is from our algorithm Algo2.

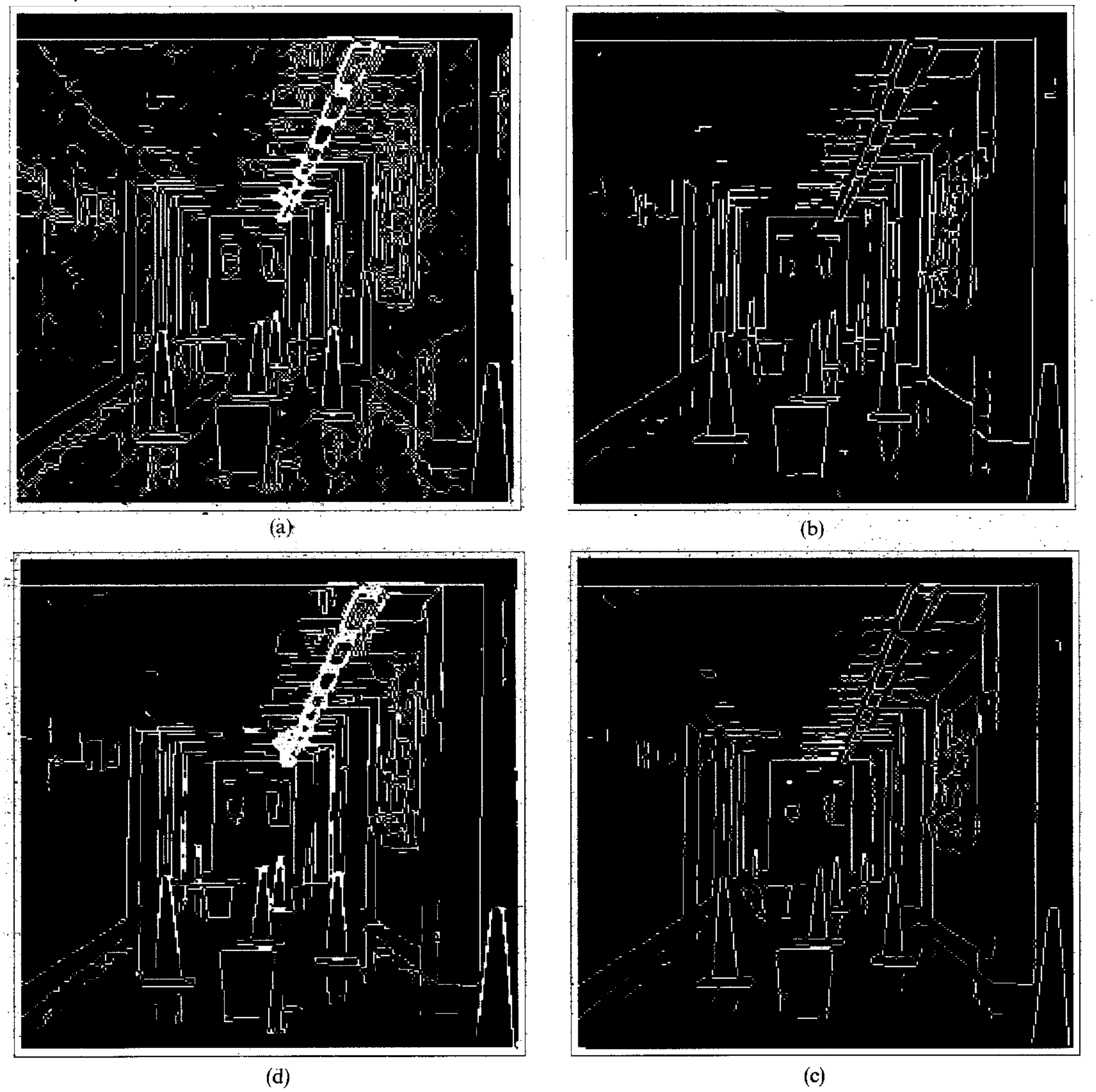


Fig. 11. An indoor scene. (a) Is from Kittler and Hancock's algorithm; (b) is from Shen and Castan's algorithm; (c) is from our algorithm Algo1 and (d) is from our algorithm Algo2.

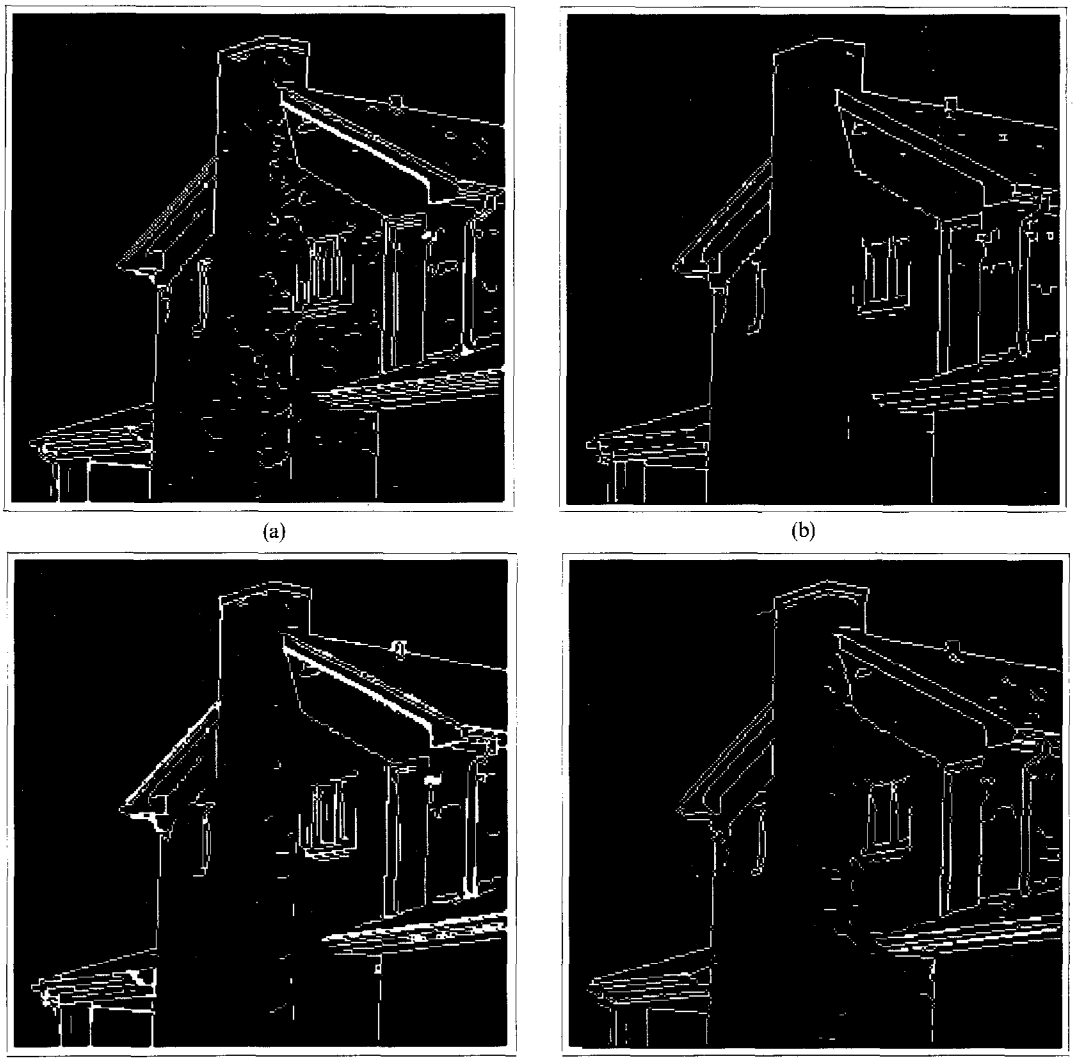
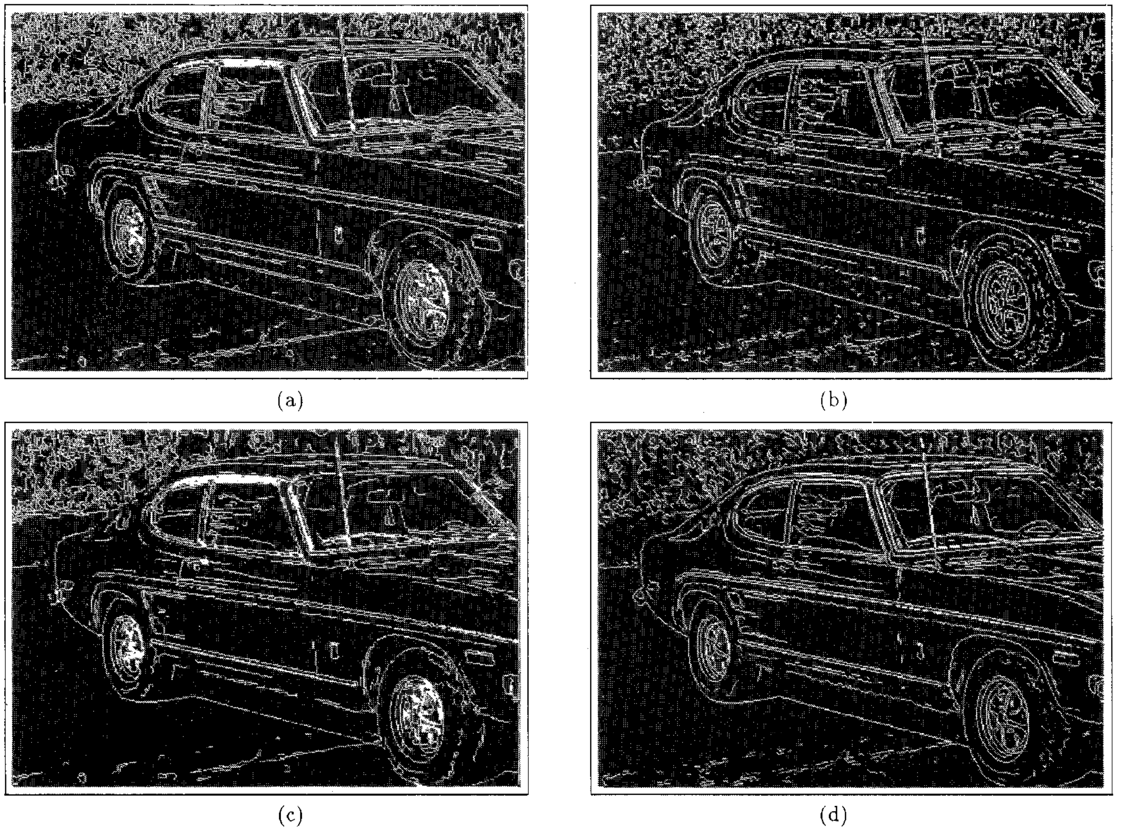


Fig. 12. The scene of a house. (a) Is from Kittler and Hancock's algorithm; (b) is from Shen and Castan's algorithm: (c) is from our algorithm Algo1 and (d) is from our algorithm Algo2.



(a) (b) (c) (d)

Fig. 13. The scene of a car. (a) is from Kittler and Hancock's algorithm; (b) is from Shen and Castan's algorithm; (c) is from our algorithm Algo1 and (d) is from our algorithm Algo2.

very important. The results from Algo2 is much better than those from Algo1;

(5) An important feature of both Kittler and Hancock's algorithm and our algorithms is therate of convergence.[31] After 10 iterations, the relaxation processes essentially converge;

5. CONCLUDING REMARKS

This paper presented an application of dictionary based relaxation label method to the edge detection problem. Throughout the paper, our main concern is how to use relaxation efficiently. Many techniques have been explored to achieve this goal. These include the dictionary model, the recursive exponential filtr, the zero-crossing of second order intensity difference and the JND concept. The experiments show that the relaxation edge detection algorithm converges quickly, and works efficiently by using contextual information to preserve edges and to eliminate noise. We found that relaxation, when use for edge detection, provides better connectedness. We also found that the quality of initial label assignment has a significant impact on the quality of the edge detection.

For further research, the relation between relaxation technique and neural network is an important issue.[28,29] has found that the general relaxation scheme and Hopfield networks are closely related. Since relaxation is a computational complex task, a study of dictionary based relaxation labeling methods and neural networks has the potential to find an efficient massive parallel implementation.

Acknowledgement—Special thanks to Mr Hla Min for proof-reading the paper.

REFERENCES

1. V. S. Nalwa and T. O. Binfold, On detecting edges, *IEEE Trans.* **PAMI-8**(6), 699–741 (1986).
2. J. Shen and S. Castin, Further results on DRF method for edge detection, *Ninth Int. Conf. on Pattern Recognition* Rome (1988).
3. L. G. Roberts, Machine perception of three-dimensional solids, *Optical and Electro-Optical Information Processing*, MIT Press, Cambridge, Massachusetts (1965).
4. I. Sobel, Camera models and machine perception, *Stanford AI MEMO* 121, Dept. of Computer Science, Standford University (1970).
5. J. Canny, Computational approach to edge detection, *IEEE Trans.* **PAMI-8**(6), 699–714 (1986).
6. R. Deriche, Optimal edge detection using recursive filtering, *Proc. First Int. Conf. on Computer Vision*, London, (8 June 1987).

7. J. Shen and S. Castin, An optimal linear operator for edge detection, *Proc. SPIE*, **87**, Miami (1987).
8. E. R. Hancock, Resolving edge-line ambiguities using probabilistic relaxation, *CVPR* 93, 300–306 (1993).
9. R. Mehrotra, K. R. Namuduri and N. Ranganathan, Gabor filter-based edge detection, *Pattern Recognition* **25**, 1479–1493 (1992).
10. M. Nitzberg and T. Shiota, Nonlinear image filtering with edge and corner enhancement, *IEEE Trans.* **PAMI-14**(8), 826–833 (1992).
11. P. Perona and J. Malik, Scale-space and edge detection using anisotropic diffusion, *IEEE Trans.* **PAMI-12**(7), 629–639 (1990).
12. Y. Wu, S. S. Iyengar and H. Min, Efficient edge extraction of images by directional tracing, Technical Report, Dept of Computer Science, Louisiana State University (1993).
13. D. O. Ungureanu *et al.*, A real time edge linker, *CVPR* **93**, 793–794 (1993).
14. J. S. Chen and G. Medioni, Detection, localization and estimation of edges, *IEEE Trans.* **PAMI-11**(2), 191–198 (1989).
15. R. M. Haralick and J. S. J. Lee, Context dependent edge detection, *Proc. Ninth Int. Conf. on Pattern Recognition* Rome, 203–207 (1989).
16. W. E. Higgins and C. M. Hsu, Edge detection using 2D locel structural information, *ICAPPS* 91, 2549–2552 (1991).
17. Y. Wang and S. K. Mitra, Edge detection based on orientation distribution of gradinet images, *ICAPPS* 91, 2569–2572 (1991).
18. A. Rosenfeld, R. A. Hummel and S. W. Zucker, Scene labeling by relaxation operations, *IEEE Trans.* **SMC-6**, 420–433 (1976).
19. S. Geman and D. Geman, Stochastic relaxation, Gibbs distributions and Bayesian restoration of images, *IEEE* **PAMI-6**, 721–741 (1984).
20. L. S. Davis, C. Y. Wang and H. C. Xie, An experiment in multispectral, multitemporal crop classification using relaxation techniques, *Comput. Vision, Graph. Image Process*, **23**, 227–235 (1983).
21. E. R. Hancock and J. Kittler, Edge-labeling using dictionary-based relaxation, *IEEE Trans.* **PAMI-12**(12), 165–181 (1990).
22. F. R. Hansen and H. Elliot, Image segmentation using simple Markov Random fields, *Computer Graphics Image Process.* **20**, 101–132 (1982).
23. B. J. Schachter, A. Lev, S. W. Zucker and A. Rosenfeld, An application of relaxation methods to edge reinforcement, *IEEE Trans.* **SMC**, **7**, 813–816 (1977).
24. E. R. Hancock and J. Kittler, Discrete relaxation, *Pattern Recognition* **23**, (7), 711–733 (1990).
25. E. R. Hancock and J. Kittler, Adaptive estimation of Hysteresis thresholds, *IEEE CVPR* 91, 196–201 (1991).
26. J. Kittler and E. R. Hancock, Combining evidence in probabilistic relaxation, *Int. J. Pattern Recognition Artif. Intell.* **3**, 29–52 (1989).
27. W. Deng and S. S. Iyengar, A new probability relaxation scheme and its application to edge detection, submitted.
28. S. Yu and W. Tsai, Relaxation by the hopfield neural network, *Pattern Recognition* **25**, 197–209 (1992).
29. C. T. Genis, Relaxation and neural learning: points of convergence and divergence, *J. Parallel Distr. Comput.* **6**, 217–244 (1989).
30. L. Pelkowitz, A continuous relaxation labeling algorithm for Markov Random fields, *IEEE Trans. SMC* **20**, 709–715 (1990).
31. J. Kittler and J. Llingworth, Relaxation labeling algorithms—a review, *Image and Vision Comput.* **3**, 206–216 (1985).

About the Author—S. SITHARAMA IYENGAR (M'88–SM' 90 under contract N00014-92-J-6003) is the chairman of the Computer Science Department and Professor of Computer Science at Louisiana State University. He has directed LSU's Robotics Research Laboratory since its inception in 1986. He has been actively involved with research in high-performance algorithms and data structures since receiving the Ph.D. degree in 1974, and has directed more than 22 Ph.D. dissertations at LSU. He has served as principal investigator on research projects supported by the Office of Naval Research, the National Aeronautics and Space Administration, the National Science Foundation/Laser Program, the California Institute of Technology's Jet Propulsion Laboratory, the Department of Navy-NORDA, the Department of Energy (through Oak Ridge National Laboratory, Tennessee), the LEQFS-Board of Regents, and Apple Computers. He has edited a two-volume tutorial on autonomous Mobile Robots and has edited two other books and more than 160 publications—including 90 archival journal papers in areas of high-performance parallel and distributed algorithms and data structure for image processing and pattern recognition, autonomous navigation, and distributed sensor networks. Iyengar was a visiting professor (fellow) at JPL, the Oak Ridge National Laboratory, and the Indian Institute of Science. He is also an Association for Computing Machinery national lecturer, a series editor for *Neuro Computing of Complex Systems*, and area editor for the *Journal of Computer Science and Information.* He has served as guest editor for the *IEEE Transactions on Software Engineering* (1988); *Computer magazine* (1989); *the IEEE Transactions on System, Man and Cybernetics*; the *IEEE Transactions on Knowledge and Data Engineering*; and the *Journal of Computers and Electrical Engineering.*

Dr Iyengar was awarded the Phi Delta Kappa Research Award of Distinction at LSU in 1989, won the Best Teacher Award in 1978, and received the Williams Evans Fellowship from the University of Otago, New Zealand, in 1991.

About the Author—WEIAN DENG was born on the 30th of January, 1963, in Zhaoqing, Guangdong, China. He graduated with a B.E. in Computer Science and Engineering from Beijing University of Aeronautics and Astronautics, Beijing, China in 1984 and a ME in Computer Science from Zhongshan University of Computer Science from July, 1987 to August, 1990. He has a Ph.D. in Computer Science gained at Louisiana State University. He is now an assistant professor in the Computer Science Program at Teikyo Westmar University. He is interested in Image Processing, Computer Vision, Parallel Algorithm, Neural Networks, Natural Language Processing and Artificial Intelligence.