Pattern Recognition, Vol. 23, No. 1/2, pp. 1–19, 1990
Printed in Great Britain.

0031-3203/90 $3.00 + .00
Pergamon Press plc
Pattern Recognition Society

CONTEXT DEPENDENT EDGE DETECTION AND EVALUATION

ROBERT M. HARALICK* and JAMES S. J. LEE†
*EE Department, FT-10, University of Washington, Seattle, WA 98195, U.S.A.; and †Neo Path Inc., Seattle, WA 98122, U.S.A.

(*Received* 30 *May* 1986; *in revised form* 8 *March* 1988; *received for publication* 8 *February* 1989)

Abstract—To simulate the edge perception ability of human eyes and detect scene edges from an image, context information must be employed in the edge detection process. To accomplish the optimal use of context, we introduce an edge detection scheme which uses the context of the whole image. The edge context for each pixel is the set of all row monotonically increasing paths through the pixel. The edge detector assigns a pixel that edge state having highest edge probability among all the paths.

Based on the same framework, we developed a general robust evaluator for edge detectors. The scheme, based on local edge coherence, does not require any prior information about the ideal edge image and allows any size of neighborhood with which local edge coherence is evaluated on the basis of continuity, thinness, and positional accuracy. The edge evaluator can be incorporated with a feedback mechanism to automatically adjust edge detection parameters (e.g. edge thresholds), for adaptive detection of edges in real images.

Experiments indicate the validity of the edge detector and the general edge evaluator. Upon comparing the performance of the context dependent edge detector with the context free second directional derivative zero-crossing edge operator, we find that the context dependent edge detector is superior.

Edge detection Context Markov dependence Bayes

1. INTRODUCTION

Edges in a scene are the consequence of changes in some physical and surface properties, such as illumination (shadows, for example), geometry (orientation or depth) and reflectance. As there is a direct relationship between the edges and physical properties of a scene, much of the scene information can be recovered from an edge image. Thus, edge detection plays a key step in the early processing of a computer vision system.

Image edges occur in places of significant intensity changes on the image. There are many kinds of intensity changes in an image. The usual aim of edge detection is to locate edges belonging to boundaries of objects of interest. While the human eyes perform this task easily, the detection of edges is a complex task to achieve. The difficulties in edge detection are mostly caused by noise, blurring and quantizing effects. This results in a situation in which not all the image edges correspond to scene edges and vice versa. People incorporating world knowledge and contextual information can detect edges selectively. Minor image edges which do not correspond to main scene edges are ignored and major scene edges are detected even though they do not correspond perfectly to image edges.

A variety of edge detection schemes have been proposed in the past decade. Most of these operators perform reasonably well on simple noise free images whereas they tend to fail on the images degraded by noise. This is because that as the noise of an image increases, the correspondence between image edges and scene edges becomes weaker and weaker. Thus, an edge detector which can perform well on noisy images is most desirable.

Even though it is possible to derive an edge detection algorithm and argue that it is mathematically optimal under certain ideal image models [Marr and Hildreth 14; Canny 2], any detector which is optimal for an image intensity edge is not necessarily optimal for a scene edge. One possible solution to this exploits the fact that the edges in the real scene domain are not difficult to define and they are less ambiguous than the edges defined in the image domain. The ambiguity with image edges is mainly caused by the unknown factors involved in the many to one physics governing image acquisition.

The solution to edge detection on noisy images should not be image smoothing, because image smoothing alone tends to blur edges. The best solution, we believe, is to incorporate world knowledge and edge context information into the edge detection process.

The context approach described here is related to the dynamic programming idea of Montanari [16] and Martelli [15] of linking together edge segments. However, the dynamic programming, as they employed it is basically a postprocessing process whose performance heavily relies on the starting points for linking which are provided by the preprocessing. A context dependent edge detection using relaxation labeling was described by Zucker *et al.* [19]. Their scheme is more computationally expensive than ours and does not have a true probability interpretation.

It is of interest to evaluate the quality of an edge detector, both to compare one detection scheme against another, and also to study the behavior of a given detector under different conditions and parameter settings. Several authors have proposed techniques for edge evaluation [Abdou and Pratt 1; Fram and Deutsch 4; Peli and Malah 18]. Most of these schemes require prior knowledge of the true edge position and lack a continuity measure. None of them exploit consistency in the direction of the detected edges. Kitchen and Rosenfeld [12] developed a fully automatic edge evaluation technique based on the idea of local edge coherence. This scheme has an inherent bias against curved edges and it only allows edge lines a single pixel wide and measures edge coherence within a 3×3 neighborhood. It also disregards the correct location of the edges. Thus, an edge detector that systematically mislocates edges will receive an evaluation measure equal to that of a detector which perfectly locates edges. Additionally, the approach is basically ad hoc, with no underlying theory.

In this paper, based on the same framework as the context edge detection scheme, we formulate the edge evaluation problem as a Bayesian decision problem and show that the edge evaluation of Kitchen and Rosenfeld [12] is just a special case of our general solution. We show how the edge evaluator measures edge coherence from any size of neighborhood as well as the correctness of edge position. This is the first time a general edge position correctness measure has been disclosed.

2. CONTEXT DEPENDENT EDGE DETECTION

2. *Edge context*

To explain the meaning of the edge context of an image, fix attention on any pixel in the image. Now consider all the row monotonically increasing paths which begin at any border pixel of the image above the selected pixel, go through the selected pixel, and end at some border pixel of the image below the selected pixel. Each such path represents a context for the pixel. Corresponding to each path and the observed pixel values on the path, there is an associated probability of edge state for the given pixel. Among all the paths there is some best 'edge' path which assigns the current pixel as an 'edge' pixel with a probability that is higher than the probability of every other 'edge' path. In general it is not necessary that all the pixels in an 'edge' path be 'edge' pixels. In the following derivations we allow 'no-edge' pixels in an edge path. However, the derivations are as general as that by a minor modification, we can require all the edge path pixels to have edge state 'edge'.

Similarly, there is some best 'no-edge' path which assigns the current pixel as a 'no-edge' pixel with a probability that is higher than the probability of every other 'no-edge' path. In considering the difference between the edge and no-edge context, we do not use the best 'no-edge' path alone for the no-edge context. For a given pixel (r, c), a row monotonically increasing path has to pass through one of the pixels $(r-1, c-1)$, $(r-1, c)$, $(r-1, c+1)$, and $(r, c-1)$ before entering the pixel (r, c) and it has to go through one of the pixels among $(r+1, c-1)$, $(r+1, c)$, $(r+1, c+1)$, and $(r, c+1)$ when leaving (r, c). Thus, for each entering and leaving pixel pair there exists a best non-edge path. The non-edge context we use for the edge detection is the average non-edge probability of these best paths.

In Haralick and Lee [10] and Lee [13], a context dependent edge detection scheme was introduced. The context used is basically the edge data in the neighborhood of the pixel under consideration. Any pixels outside this neighborhood have nothing to do with the edge detection of the current pixel. In this paper we introduce a context dependent edge detection scheme which uses all the edge data in the image as the context to help the edge detection process.

According to Haralick and Lee [10], the edge detection problem can be formulated as a Bayesian decision problem. The solution to this problem is: for each pixel position (r, c) of the image assign the edge state ε_{rc} as 'edge' if

$$P(\varepsilon^*_{rc} = \text{'edge'} \mid K) > P(\varepsilon^*_{rc} = \text{'no} - \text{edge'} \mid K) \quad (1)$$

and assign the edge state ε_{rc} as 'no-edge' otherwise. The context information which appears in equation (1) is denoted by K which is the facet model representation of each pixel's local neighborhood [Haralick *et al.* 1980] of the whole image. Thus, we have the most desirable kind of edge labeling process: a process which labels each pixel with that edge state having the highest probability given the entire context of the image. In line with equation (1), the context dependent scheme assigns a pixel edge state 'edge' if the edge probability of the best 'edge' path is higher than the average no-edge probability of the best 'no-edge' paths and assigns a pixel 'no-edge' otherwise.

We begin our description with some definitions. The set U_{rc} designates the set of all row monotonically increasing paths which begin at some border pixel of the image above or to the left of pixel (r, c) and terminate at pixel (r, c). The set L_{rc} designates the set of all row monotonically increasing paths which begin at (r,c) and terminate at some border pixel below or to the right of pixel (r, c). These are illustrated in Figs 1(a) and (b). Let $N_1(r, c) = \{(r-1, c-1), (r-1, c), (r-1, c+1), (r, c-1)\}$, and $N_2(r, c) = \{(r+1, c-1), (r+1, c), (r+1, c+1), (r, c+1)\}$. The set $U_{rc(pq)}$ where $(p, q) \in N_1(r, c)$ is defined as

$$U_{rc(pq)} = \{T : T \in U_{rc} \text{ and } (p, q) \in T\}.$$

Similarly, we can define

$$L_{rc(ij)} = \{T: T\in L_{rc} \text{ and } (i,j)\in T\}$$

where $(i,j)\in N_2(r,c)$.

The set Z_{rc} designates the set of all row monotonically increasing paths beginning at a border of the image passing through pixel (r,c) and continuing to another border pixel of the image. The relationship between Z_{rc} and L_{rc} and U_{rc} should be obvious. Z_{rc} is just the join of all paths in U_{rc} with the paths in L_{rc}. Similarly, we can define $Z_{rc(pq,ij)}$ as the join of all paths in $U_{rc(pq)}$ with the paths in $L_{rc(ij)}$.

The set U^*_{rc} designates the set of all row monotonically increasing paths which begin at some border pixel of the image at the same row or above pixel (r,c) and terminate at pixel (r,c). The set L^*_{rc} designates the set of all row monotonically increasing paths which begin at pixel (r,c) and terminate at some border pixel at the same row or below the pixel (r,c). This is illustrated in Figs 1(c) and (d).

2.2 *Detection algorithms*

For pixel (r,c), let $\underline{k}_{rc}$ designate its facet parameter representation and $f_{Z_{rc}}(\varepsilon^*_{rc})$ be the probability that its true edge state is ε^*_{rc} given the facet parameters of the best row monotonically increasing path T taking the direction θ^*_{rc0} through (r,c), where θ^*_{rc0} is the direction which maximizes $P(\theta^*_{rc}|\underline{k}_{rc})$. Let $g_{Z_{rc}}$ be

$$g_{Z_{rc}}(e^*_{rc},\theta^*_{rc}) = \max_{T\in Z_{rc}} P(\varepsilon^*_{rc},\theta^*_{rc}|\underline{k}_{ij}:(i,j)\in T). \tag{2}$$

Then,

$$f_{Z_{rc}}(\varepsilon^*_{rc}) = g_{Z_{rc}}(\varepsilon^*_{rc},\theta^*_{rc0}).$$

It is noted that $f_{Z_{rc}}$ is not a function of θ^*_{rc0} because θ^*_{rc0} is a fixed value which is determined by

$$P(\theta^*_{rc0}|\underline{k}_{rc}) > P(\theta_{rc}|\underline{k}_{rc})\,\forall\,\theta_{rc}\neq\theta^*_{rc0}.$$

Similarly, we can define $f_{Z_{rc(pq,ij)}}(\varepsilon^*_{rc})$. The average probability $\bar{f}_{Z_{rc}}(\varepsilon^*_{rc})$is defined by

$$\bar{f}_{Z_{rc}}(\varepsilon^*_{rc}) = \frac{1}{16}\sum_{\substack{(pq)\in N_1(r,c)\\(ij)\in N_2(r,c)}} f_{Z_{rc(pq,ij)}}(\varepsilon^*_{rc}). \tag{3}$$

The Bayesian decision theory based edge detection scheme (1) can be expressed as

$$\varepsilon^*_{rc} = \begin{cases}\text{edge} & \text{if } \bar{f}_{Z_{rc}}(\varepsilon^*_{rc} = \text{'edge'}) > \bar{f}_{Z_{rc}}(\varepsilon^*_{rc} = \text{'no-edge'})\\ \text{no-edge} & \text{otherwise.}\end{cases} \tag{4}$$

To perform edge detection we have to compute $f_{Z_{rc}}$

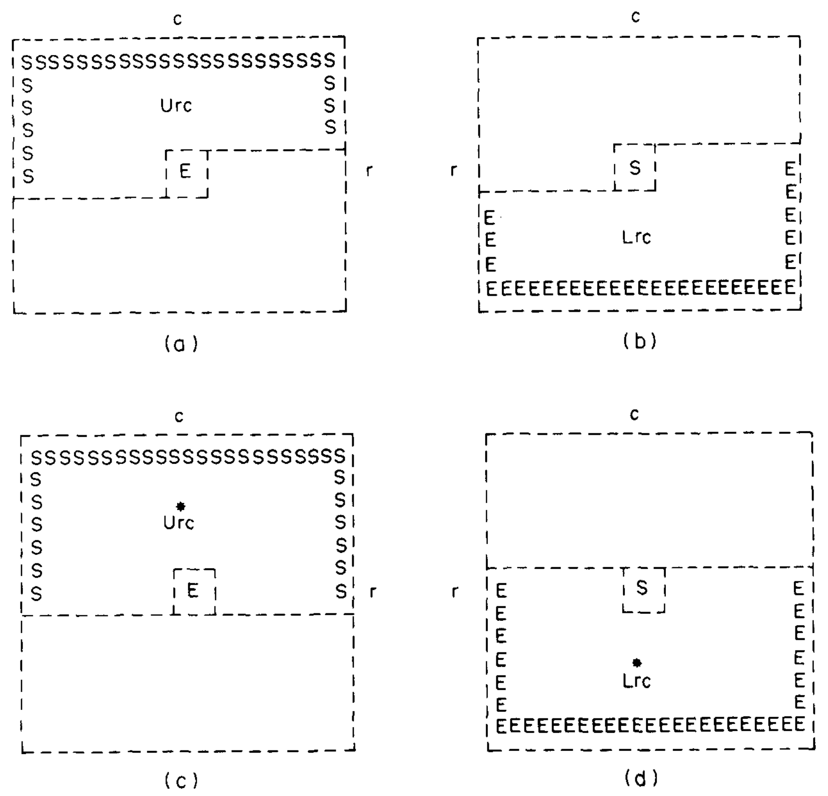

Fig. 1. Illustrates (a) the set U_{rc} and (b) the set L_{rc}. U_{rc} is the set of all row and column monotonically increasing paths beginning at a border of the image above or to the left of the pixel (r,c) and terminating at pixel (r,c). L_{rc} is the set of all row and column monotonically increasing paths beginning at the pixel (r,c) and terminating at the border of the image below or to the right of the pixel. Similarly, the sets U^*_{rc} and L^*_{rc} are illustrated in (c) and (d).

and then derive $\bar{f}_{Z_{rc}}$ from $f_{Z_{rc}}$. To compute $f_{Z_{rc}}$, we have to compute $g_{Z_{rc}}$ first.

Analogous to the definition of $g_{Z_{rc}}(\varepsilon^*_{rc},\theta^*_{rc})$, $g_{U_{rc}}(\varepsilon^*_{rc},\theta^*_{rc})$ and $g_{L_{rc}}(\varepsilon^*_{rc},\theta^*_{rc})$ are defined as follows:

$$g_{U_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}) = \max_{T\in U_{rc}} P(\varepsilon^*_{rc},\theta^*_{rc}|\underline{k}_{ij}:(i,j)\in T) \quad (5)$$

and

$$g_{L_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}) = \max_{T\in L_{rc}} P(\varepsilon^*_{rc},\theta^*_{rc}|\underline{k}_{ij}:(i,j)\in T). \quad (6)$$

$g_{U_{rc(pq)}}(\varepsilon^*_{rc},\theta^*_{rc})$ and $g_{L_{rc(ij)}}(\varepsilon^*_{rc},\theta^*_{rc})$ are defined in a similar fashion. $g_{Z_{rc}}(\varepsilon^*_{rc},\theta^*_{rc})$ can be decomposed as follows,

$$\begin{aligned} g_{Z_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}) &= \max_{T\in Z_{rc}} P(\varepsilon^*_{rc},\theta^*_{rc}|\underline{k}_{ij}:(i,j)\in T) \\ &= \max_{T\in Z_{rc}} \sum_{\substack{\theta^*_{ij},\varepsilon^*_{ij}\\(i,j)\in T^-}} P(\underline{k}_{ij}:(i,j)\in T|\varepsilon^*_{ij},\theta^*_{ij},(i,j)\in T) \\ &\quad * \frac{P(\varepsilon^*_{ij},\theta^*_{ij},(i,j)\in T)}{P(\underline{k}_{ij}:(i,j)\in T)} \end{aligned} \quad (7)$$

where T^- designates the set of all pixels in T but pixel (r,c).

It is reasonable to assume that when the pixel (r,c) is being examined, no observed characteristics from any other pixel but pixel (r,c) affects the observed data of position (r,c). Hence

$$\begin{aligned} &P(\underline{k}_{ij}:(i,j)\in T|\varepsilon^*_{ij},\theta^*_{ij},(i,j)\in T) \\ &\quad = \prod_{(i,j)\in T} P(\underline{k}_{ij}|\varepsilon^*_{ij},\theta^*_{ij},(i,j)\in T) \end{aligned} \quad (8)$$

and

$$P(\underline{k}_{ij}:(i,j)\in T) = \prod_{(i,j)\in T} P(\underline{k}_{ij}). \quad (9)$$

It is also reasonable to assume that the observed facet measurements $\underline{k}_{ij}$ at a pixel (i,j) depends only upon its true facet parameters and edge data. Hence

$$P(\underline{k}_{ij}|\underline{k}^*_{ij},\varepsilon^*_{kl},\theta^*_{kl}:\ (k,l)\in T) = P(\underline{k}_{ij}|\underline{k}^*_{ij},\varepsilon^*_{ij},\theta^*_{ij}). \quad (10)$$

Based on the above two assumptions we can derive

$$\begin{aligned} &P(\underline{k}_{ij}:(i,j)\in T|\varepsilon^*_{ij},\theta^*_{ij},(i,j)\in T) \\ &\quad = \prod_{(i,j)\in T} P(\underline{k}_{ij}|\varepsilon^*_{ij},\theta^*_{ij}). \end{aligned} \quad (11)$$

The joint probability $P(\varepsilon^*_{ij},\theta^*_{ij},(i,j)\in T)$ contains all the information about context. A second order generalized conditional independence assumption is used. It is a Markov-like assumption and is the simplest assumption of higher order than independence. Based on this assumption the joint prior probability can be expressed as the product of functions whose arguments are the label pairs for successive pixels in the path. Let $R(T)$ designate the set of all pairs of successive pixels in the path T, we have

$$P(\varepsilon^*_{ij},\theta^*_{ij}:(i,j)\in T) = \prod_{((i,j),(k,l))\in R(T)} a(\varepsilon^*_{ij},\theta^*_{ij},\varepsilon^*_{kl},\theta^*_{kl}). \quad (12)$$

Using assumptions (9), (11), and (12), (7) can be decomposed into

$$\begin{aligned} g_{Z_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}) &= \max_{T\in Z_{rc}} \sum_{\substack{\theta^*_{ij},\varepsilon^*_{ij}\\(i,j)\in T^-}} \prod_{(i,j)\in T^-} \frac{P(\underline{k}_{ij}|\varepsilon^*_{ij},\theta^*_{ij})}{P(\underline{k}_{ij})} \\ &\quad * \prod_{((i,j),(k,l))\in R(T)} a(\varepsilon^*_{ij},\theta^*_{ij},\varepsilon^*_{kl},\theta^*_{kl}). \end{aligned} \quad (13)$$

Since Z_{rc} can be decomposed into the join of U_{rc} and L_{rc}, this results in

$$\begin{aligned} g_{Z_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}) &= \max_{T_1\in U_{rc}} \max_{T_2\in L_{rc}} \sum_{\substack{\theta^*_{ij},\varepsilon^*_{ij}\\(i,j)\in T_1^-}} \sum_{\substack{\theta^*_{ij},\varepsilon^*_{ij}\\(i,j)\in T_2^-}} \\ &\prod_{(i,j)\in T_1} \frac{P(\underline{k}_{ij}|\varepsilon^*_{ij},\theta^*_{ij})}{P(\underline{k}_{ij})} \prod_{(i,j)\in_2} \frac{P(\underline{k}_{ij}|\varepsilon^*_{ij},\theta^*_{ij})}{P(\underline{k}_{ij})} * \frac{P(\underline{k}_{rc})}{P(\underline{k}_{rc}|\varepsilon^*_{rc},\theta^*_{rc})} \\ &\quad * \prod_{((i,j),(k,l))\in R(T_1)} a(\varepsilon^*_{ij},\theta^*_{ij},\varepsilon^*_{kl},\theta^*_{kl}) \\ &\quad * \prod_{((i,j),(k,l))\in R(T_2)} a(\varepsilon^*_{ij},\theta^*_{ij},\varepsilon^*_{kl},\theta^*_{kl}). \end{aligned} \quad (14)$$

Rearranging (14) we can group all expressions involving T_1 together and all expressions involving T_2 together and obtain

$$\begin{aligned} g_{Z_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}) &= \frac{P(\underline{k}_{rc})}{P(\underline{k}_{rc}|\varepsilon^*_{rc},\theta^*_{rc})} \\ &\quad * g_{U_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}) * g_{L_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}). \end{aligned} \quad (15)$$

Similarly

$$\begin{aligned} g_{Z_{rc(pq,ij)}}(\varepsilon^*_{rc},\theta^*_{rc}) &= \frac{P(\underline{k}_{rc})}{P(\underline{k}_{rc}|\varepsilon^*_{rc},\theta^*_{rc})} g_{U_{rc(pq)}}(\varepsilon^*_{rc},\theta^*_{rc}) \\ &\quad * g_{L_{rc(ij)}}(\varepsilon^*_{rc},\theta^*_{rc}). \end{aligned}$$

The decomposition of $g_{U_{rc}}$ is in terms of the neighboring $g_{U_{rc-1}}$, $h_{U^*_{r-1,c-1}}$, $h_{U^*_{r-1,c}}$, and $h_{U^*_{r-1,c+1}}$, all of which need definition. Just as the derivation in (7),

$$\begin{aligned} g_{U_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}) &= \max_{T\in U_{rc}} P(\varepsilon^*_{rc},\theta^*_{rc}|\underline{k}_{ij}:(i,j)\in T) \\ &= \max_{T\in U_{rc}} \sum_{\substack{\theta^*_{ij},\varepsilon^*_{ij}\\(i,j)\in T^-}} \prod_{(i,j)\in T} \frac{P(\underline{k}_{ij}|\varepsilon^*_{ij},\theta^*_{ij})}{P(\underline{k}_{ij})} \\ &\quad * \prod_{((i,j),(k,l))\in R(T)} a(\varepsilon^*_{ij},\theta^*_{ij},\varepsilon^*_{kl},\theta^*_{kl}) \end{aligned} \quad (16)$$

and

$$\begin{aligned} h_{U^*_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}) &= \max_{T\in U^*_{rc}} P(\varepsilon^*_{rc},\theta^*_{rc}|\underline{k}_{ij}:(i,j)\in T) \\ &= \max_{T\in U^*_{rc}} \sum_{\substack{\theta^*_{ij},\varepsilon^*_{ij}\\(i,j)\in T^-}} \prod_{(i,j)\in T} \frac{P(\underline{k}_{ij}|\varepsilon^*_{ij},\theta^*_{ij})}{P(\underline{k}_{ij})} \\ &\quad * \prod_{((i,j),(k,l))\in R(T)} a(\varepsilon^*_{ij},\theta^*_{ij},\varepsilon^*_{kl},\theta^*_{kl}). \end{aligned} \quad (17)$$

To do the decomposition for $g_{U_{rc}}$ we need to recognize that whatever the best path is, the best path to (r, c) must have come from one of the pixel locations: $(r, c-1)$, $(r-1, c-1)$, $(r-1, c)$, or $(r-1, c+1)$. Because the best paths cannot cross itself, if the best path came from $(r, c-1)$, then the path must be in $U_{r,c-1}$. However, there is no danger of the path crossing itself if the best path comes from $(r-1, c-1)$. Hence, such a path must be in $U^*_{r-1,c-1}$. Likewise, a best path coming from $(r-1, c)$ must be in $U^*_{r-1,c}$ and a best path coming from $(r-1, c+1)$ must be in $U^*_{r-1,c+1}$. Using this idea, we can express (16) as

$$
\begin{aligned}
g_{U_{rc}}(\varepsilon^*_{rc}, \theta^*_{rc}) = {} & \frac{P(\underline{k}_{rc} | \varepsilon^*_{rc}, \theta^*_{rc})}{P(\underline{k}_{rc})} \\
& * \max \Bigg\{ \max_{T \in U_{r,c-1}} \sum_{\substack{\theta^*_{ij}, \varepsilon^*_{ij} \\ (i,j)\in T}} \prod_{(i,j)\in T} \frac{P(\underline{k}_{ij} | \varepsilon^*_{ij}, \theta^*_{ij})}{P(\underline{k}_{ij})} \\
& * \prod_{((i,j),(k,l))\in R(T)} a(\varepsilon^*_{ij}, \theta^*_{ij}, \varepsilon^*_{kl}, \theta^*_{kl}) \\
& * (a(\varepsilon^*_{r,c-1}, \theta^*_{r,c-1}, \varepsilon^*_{rc}, \theta^*_{rc}), \\
& \max_{T \in U^*_{r-1,c-1}} \sum_{\substack{\theta^*_{ij}, \varepsilon^*_{ij} \\ (i,j)\in T}} \prod_{(i,j)\in T} \frac{P(\underline{k}_{ij} | \varepsilon^*_{ij}, \theta^*_{ij})}{P(\underline{k}_{ij})} \\
& * \prod_{((i,j),(k,l))\in R(T)} a(\varepsilon^*_{ij}, \theta^*_{ij}, \varepsilon^*_{kl}, \theta^*_{kl}) \\
& \qquad * a(\varepsilon^*_{r-1,c-1}, \theta^*_{r-1,c-1}, \varepsilon^*_{rc}, \theta^*_{rc}), \\
& \max_{T \in U^*_{r-1,c}} \sum_{\substack{\theta^*_{ij}, \varepsilon^*_{ij} \\ (i,j)\in T}} \prod_{(i,j)\in T} \frac{P(\underline{k}_{ij} | \varepsilon^*_{ij}, \theta^*_{ij})}{P(\underline{k}_{ij})} \\
& * \prod_{((i,j),(k,l))\in R(T)} a(\varepsilon^*_{ij}, \theta^*_{ij}, \varepsilon^*_{kl}, \theta^*_{kl}) \\
& * a(\varepsilon^*_{r-1,c}, \theta^*_{r-1,c}, \varepsilon^*_{rc}, \theta^*_{rc}), \\
& \max_{T \in U^*_{r-1,c+1}} \sum_{\substack{\theta^*_{ij}, \varepsilon^*_{ij} \\ (i,j)\in T}} \prod_{(i,j)\in T} \frac{P(\underline{k}_{ij} | \varepsilon^*_{ij}, \theta^*_{ij})}{P(\underline{k}_{ij})} \\
& * \prod_{((i,j),(k,l))\in R(T)} a(\varepsilon^*_{ij}, \theta^*_{ij}, \varepsilon^*_{kl}, \theta^*_{kl}) \\
& * a(\varepsilon^*_{r-1,c+1}, \theta^*_{r-1,c+1}, \varepsilon^*_{rc}, \theta^*_{rc}) \Bigg\}. \qquad (18)
\end{aligned}
$$

When examining the first term in the maximization and comparing it to the term $g_{U_{r,c-1}}(\varepsilon^*_{r,c-1}, \theta^*_{r,c-1})$ defined by (16) we discover that the expressions are almost identical. The only difference is that the expression for $g_{U_{r,c-1}}(\varepsilon^*_{r,c-1}, \theta^*_{r,c-1})$ involves a summation iterated over all $(i,j) \in T^-$ while the first expression in the maximization involves a summation iterated over all $(i,j) \in T$. Since the maximization done in the first term is over all paths terminating at $(r, c-1)$, the summation over $\varepsilon^*_{r,c-1}$ and $\theta^*_{r,c-1}$ can be interchanged with the maximization over all $T \in U_{r,c-1}$. A similar reorganization can be done with the second, third, and fourth terms of the summation after comparison of them with $h_{U^*_{r-1,c-1}}$, $h_{U^*_{r-1,c}}$, $h_{U^*_{r-1,c+1}}$. This results in

$$
\begin{aligned}
g_{U_{rc}}(\varepsilon^*_{r,c}, \theta^*_{rc}) = {} & \frac{P(\underline{k}_{rc} | \varepsilon^*_{rc}, \theta^*_{rc})}{P(\underline{k}_{rc})} \\
& * \max \Bigg\{ \sum_{\theta^*_{r,c-1}, \varepsilon^*_{r,c-1}} g_{U_{r,c-1}}(\varepsilon^*_{r,c-1}, \theta^*_{r,c-1}) \\
& * a(\varepsilon^*_{r,c-1}, \theta^*_{r,c-1}, \varepsilon^*_{rc}, \theta^*_{rc}), \\
& \sum_{\theta^*_{r-1,c-1}, \varepsilon^*_{r-1,c-1}} h_{U^*_{r-1,c-1}}(\varepsilon^*_{r-1,c-1}, \theta^*_{r-1,c-1}) \\
& * a(\varepsilon^*_{r-1,c-1}, \theta^*_{r-1,c-1}, \varepsilon^*_{rc}, \theta^*_{rc}), \\
& \sum_{\theta^*_{r-1,c}, \varepsilon^*_{r-1,c}} h_{U^*_{r-1,c}}(\varepsilon^*_{r-1,c}, \theta^*_{r-1,c}) \\
& * a(\varepsilon^*_{r-1,c}, \theta^*_{r-1,c}, \varepsilon^*_{rc}, \theta^*_{rc}), \\
& \sum_{\theta^*_{r-1,c+1}, \varepsilon^*_{r-1,c+1}} h_{U^*_{r-1,c+1}}(\varepsilon^*_{r-1,c+1}, \theta^*_{r-1,c+1}) \\
& * a(\varepsilon^*_{r-1,c+1}, \theta^*_{r-1,c+1}, \varepsilon^*_{rc}, \theta^*_{rc}) \Bigg\}. \qquad (19)
\end{aligned}
$$

Equation (19) says that for each edge label $\varepsilon^*_{rc}, \theta^*_{rc}$, the conditional probability of $\varepsilon^*_{rc}, \theta^*_{rc}$ given $\{\underline{k}_{ij}: (i,j) \in T\}$ of the best path T can be obtained on the basis of the previously computed $g_{U_{r,c-1}}$, and on the previously computed $h_{U^*_{r-1,c-1}}$, $h_{U^*_{r-1,c}}$, $h_{U^*_{r-1,c+1}}$ coming from the row above the current row. Equation (19) specifies a recursive neighborhood operator which scans the image in a top down left right scan to produce for each pixel (r, c) and for each edge label $\varepsilon^*_{rc}, \theta^*_{rc}$ the probability $g_{U_{rc}}(\varepsilon^*_{rc}, \theta^*_{rc})$ providing we can demonstrate a way to compute $h_{U^*_{rc}}$. Similarly,

$$
\begin{aligned}
g_{U_{rc(r,c-1)}}(\varepsilon^*_{rc}, \theta^*_{rc}) = {} & \frac{P(\underline{k}_{rc} | \varepsilon^*_{rc}, \theta^*_{rc})}{P(\underline{k}_{rc})} \\
& * \sum_{\theta^*_{r,c-1}, \varepsilon^*_{r,c-1}} g_{U_{r,c-1}}(\varepsilon^*_{r,c-1}, \theta^*_{r,c-1}) \\
& * a(\varepsilon^*_{r,c-1}, \theta^*_{r,c-1}, \varepsilon^*_{rc}, \theta^*_{rc})
\end{aligned}
$$

and

$$
\begin{aligned}
g_{U_{rc(ij)}}(\varepsilon^*_{rc}, \theta^*_{rc}) = {} & \frac{P(\underline{k}_{rc} | \varepsilon^*_{rc}, \theta^*_{rc})}{P(\underline{k}_{rc})} \\
& * \sum_{\theta^*_{i,j}, \varepsilon^*_{i,j}} h_{U^*_{i,j}}(\varepsilon^*_{ij}, \theta^*_{ij}) \\
& * a(\varepsilon^*_{ij}, \theta^*_{ij}, \varepsilon^*_{rc}, \theta^*_{rc}),
\end{aligned}
$$

where $(i,j) \in N_1(r,c) - (r, c-1)$.

The algorithm for computing $h_{U^*_{rc}}$ is similar to that of $g_{U_{rc}}$. For a path from U^*_{rc} to reach (r, c) it must first have gone through one of the pixel location: $(r, c-1)$, $(r, c+1)$, $(r-1, c-1)$, $(r-1, c)$, or $(r-1, c+1)$. Furthermore, if it went through $(r, c-1)$, since the path cannot cross itself, it must be a path in $U_{r,c-1}$. From our development of equation (19), we know that the maximization over the probability of the paths go through $(r, c-1)$, $(r-1, c-1)$, $(r-1, c)$, and $(r-1, c+1)$ yields $g_{U_{rc}}(\varepsilon^*_{rc}, \theta^*_{rc})$. Thus we have

$$
h_{U^*_{rc}}(\varepsilon^*_{rc}, \theta^*_{rc}) = \max \Big\{ g_{U_{rc}}(\varepsilon^*_{rc}, \theta^*_{rc}),
$$

$$\frac{P(\underline{k}_{rc}|\varepsilon^*_{rc},\theta^*_{rc})}{P(\underline{k}_{rc})} * \max_{T\in U^*_{r,c+1}} \sum_{\substack{\theta^*_{ij},\varepsilon^*_{ij}\\(i,j)\in T}} \prod_{(i,j)\in T} \frac{P(\underline{k}_{ij}|\varepsilon^*_{ij},\theta^*_{ij})}{P(\underline{k}_{ij})} * \prod_{((i,j),(k,l))\in R(T)} a(\varepsilon^*_{ij},\theta^*_{ij},\varepsilon^*_{kl},\theta^*_{kl}) * a(\varepsilon^*_{r,c+1},\theta^*_{r,c+1},\varepsilon^*_{rc},\theta^*_{rc})\Bigg\}. \quad (20)$$

In a similar manner as in the development of (19), we interchange the order of the summation over $\varepsilon^*_{r,c+1}, \theta^*_{r,c+1}$ with the maximization over all path T in $U^*_{r,c+1}$. (20) becomes

$$h_{U^*_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}) = \max\Bigg\{g_{U_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}), \frac{P(\underline{k}_{rc}|\varepsilon^*_{rc},\theta^*_{rc})}{P(\underline{k}_{rc})} * \sum_{\theta^*_{r,c+1},\varepsilon^*_{r,c+1}} \max_{T\in U^*_{r,c+1}} \sum_{\substack{\theta^*_{ij},\varepsilon^*_{ij}\\(i,j)\in T^-}} \prod_{(i,j)\in T} \frac{P(\underline{k}_{ij}|\varepsilon^*_{ij},\theta^*_{ij})}{P(\underline{k}_{ij})} * \prod_{((i,j),(k,l))\in R(T)} a(\varepsilon^*_{ij},\theta^*_{ij},\varepsilon^*_{kl},\theta^*_{kl}) * a(\varepsilon^*_{r,c+1},\theta^*_{r,c+1},\varepsilon^*_{rc},\theta^*_{rc})\Bigg\}. \quad (21)$$

By the definition of (17), the bracketed expression of equation (21) is precisely $h_{U^*_{r,c+1}}(\varepsilon^*_{r,c+1},\theta^*_{r,c+1})$. This results in

$$h_{U^*_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}) = \max\Bigg\{g_{U_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}), \frac{P(\underline{k}_{rc}|\varepsilon^*_{rc},\theta^*_{rc})}{P(\underline{k}_{rc})} * \sum_{\theta^*_{r,c+1},\varepsilon^*_{r,c+1}} h_{U^*_{r,c+1}}(\varepsilon^*_{r,c+1},\theta^*_{r,c+1}) * a(\varepsilon^*_{r,c+1},\theta^*_{r,c+1},\varepsilon^*_{rc},\theta^*_{rc})\Bigg\}. \quad (22)$$

Equation (22) states that $h_{U^*_{rc}}$ can be recursively computed from $g_{U^*_{rc}}$ and the previous $h_{U^*_{r,c+1}}$ in a right–left scan of a row done after $g_{U^*_{rc}}$ has been computed. To start the recursive calculation (22), we take $h_{U^*_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}) = g_{U_{rc}}(\varepsilon^*_{rc},\theta^*_{rc})$ for that column position c which is the right-most position.

In summary, equations (19) and (22) imply the following algorithm for the computation of $g_{U_{rc}}(\varepsilon^*_{rc},\theta^*_{rc})$. From (19), we perform a top down left–right scan of the image recursively computing $g_{U_{rc}}$ from $g_{U_{r,c-1}}$, $h_{U^*_{r-1,c-1}}$, $h_{U^*_{r-1,c}}$, and $h_{U^*_{r-1,c+1}}$ which had been computed. Following the computation of $g_{U_{rc}}$ for all pixels on row e, we perform a right–left scan of row r using equation (22) to compute $h_{U^*_{rc}}$.

An absolutely mirror image derivation applies to $g_{L_{rc}}$. Namely,

$$g_{L_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}) = \frac{P(\underline{k}_{rc}|\varepsilon^*_{rc},\theta^*_{rc})}{P(\underline{k}_{rc})} * \max\Bigg\{\sum_{\theta^*_{r,c+1},\varepsilon^*_{r,c+1}} g_{L_{r,c+1}}(\varepsilon^*_{r,c+1},\theta^*_{r,c+1}) * a(\varepsilon^*_{r,c+1},\theta^*_{r,c+1},\varepsilon^*_{rc},\theta^*_{rc}), \sum_{\theta^*_{r+1,c+1},\varepsilon^*_{r+1,c+1}} h_{L^*_{r+1,c+1}}(\varepsilon^*_{r+1,c+1},\theta^*_{r+1,c+1}) * a(\varepsilon^*_{r+1,c+1},\theta^*_{r+1,c+1},\varepsilon^*_{rc},\theta^*_{rc}), \sum_{\theta^*_{r+1,c},\varepsilon^*_{r+1,c}} h_{L^*_{r+1,c}}(\varepsilon^*_{r+1,c},\theta^*_{r+1,c}) * a(\varepsilon^*_{r+1,c},\theta^*_{r+1,c},\varepsilon^*_{rc},\theta^*_{rc}), \sum_{\theta^*_{r+1,c-1},\varepsilon^*_{r+1,c-1}} h_{L^*_{r+1,c-1}}(\varepsilon^*_{r+1,c-1},\theta^*_{r+1,c-1}) * a(\varepsilon^*_{r+1,c-1},\theta^*_{r+1,c-1},\varepsilon^*_{rc},\theta^*_{rc})\Bigg\} \quad (23)$$

and

$$h_{L^*_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}) = \max\Bigg\{g_{L_{rc}}(\varepsilon^*_{rc},\theta^*_{rc}), \frac{P(\underline{k}_{rc}|\varepsilon^*_{rc},\theta^*_{rc})}{P(\underline{k}_{rc})} * \sum_{\theta^*_{r,c-1},\varepsilon^*_{r,c-1}} h_{L^*_{r,c-1}}(\varepsilon^*_{r,c-1},\theta^*_{r,c-1}) * a(\varepsilon^*_{r,c-1},\theta^*_{r,c-1},\varepsilon^*_{rc},\theta^*_{rc})\Bigg\}. \quad (24)$$

It is also easy to show that

$$g_{L_{rc(r,c+1)}}(\varepsilon^*_{rc},\theta^*_{rc}) = \frac{P(\underline{k}_{rc}|\varepsilon^*_{rc},\theta^*_{rc})}{P(\underline{k}_{rc})} * \sum_{\theta^*_{r,c+1},\varepsilon^*_{r,c+1}} g_{L_{r,c+1}}(\varepsilon^*_{r,c+1},\theta^*_{r,c+1}) * a(\varepsilon^*_{r,c+1},\theta^*_{r,c+1},\varepsilon^*_{rc},\theta^*_{rc})$$

and

$$g_{L_{rc(ij)}}(\varepsilon^*_{rc},\theta^*_{rc}) = \frac{P(\underline{k}_{rc}|\varepsilon^*_{rc},\theta^*_{rc})}{P(\underline{k}_{rc})} * \sum_{\theta^*_{i,j},\varepsilon^*_{i,j}} h_{L^*_{i,j}}(\varepsilon^*_{i,j},\theta^*_{i,j}) * a(\varepsilon^*_{i,j},\theta^*_{i,j},\varepsilon^*_{rc},\theta^*_{rc}),$$

where $(i,j)\in N_2(r,c) - (r,c+1)$.

$g_{L_{rc}}$ can be computed by a bottom up right–left scan of the image recursively from $g_{L_{r,c+1}}$, $h_{L^*_{r+1,c-1}}$, $h_{L^*_{r+1,c}}$, and $h_{L^*_{r+1,c+1}}$ which had been computed. A left–right scan of row r is then performed to compute $h_{L^*_{rc}}$.

As soon as $g_{L_{rc}}(\varepsilon^*_{rc},\theta^*_{rc})$ has been computed, it can be combined with $g_{U_{rc}}(\varepsilon^*_{rc},\theta^*_{rc})$ to compute $g_{Z_{rc}}(\varepsilon^*_{rc},\theta^*_{rc})$ (see [15]). In practice, the only useful θ^*_{rc} for edge detection is θ^*_{rc0} which maximizes $P(\theta^*_{rc}|\underline{k}_{rc})$. Hence, we determine θ^*_{rc0} first and then only compute $g_{rc}(\varepsilon^*_{rc},\theta^*_{rc0})$ for both ε^*_{rc} = 'edge' and 'no-edge'. The two probability terms $f_{Z_{rc}}(\varepsilon^*_{rc})$ and $\bar{f}_{Z_{rc}}(\varepsilon^*_{rc})$ are then readily available. The edge state of each pixel (r,c) can now be labeled by means of the rule of equation (4).

2.3. *Probabilities*

To perform the recursive algorithms (19), (22), (23), (24), we need the probability ratio $(P(k_0|\varepsilon_0^*,\theta_0^*))/(P(\underline{k}_0))$ and the edge consistency function $a(\varepsilon_0^*,\theta_0^*,\varepsilon_1^*,\theta_1^*)$ where 0 and 1 designate any adjacent pixel pairs in the image.

In Lee [13], we have derived the conditional probability as

$$P(k_2,\ldots,k_{10}|\varepsilon^*,\theta^*) = P(k_2^*,k_3^*|\varepsilon^*,\theta^*) * P(A,B|\varepsilon^*,\theta^*) * P(k_4,\ldots,k_{10}|A,B) \quad (25)$$

where

$$A = 6[k_7 \sin^3\theta + k_8 \sin^2\theta\cos\theta + k_9\sin\theta\cos^2\theta + k_{10}\cos^3\theta]$$

$$B = 2[k_4\sin^2\theta + k_5\sin\theta\cos\theta + k_6\cos^2\theta].$$

Let λ be the edge to no-edge ratio of K_2 standard deviation. Then $P(k_2,k_3|\varepsilon^* = \text{'edge'},\theta^*)$ can be described by a normal distribution

$$N\left(H\begin{pmatrix} 0 \\ \sqrt{\frac{\pi}{2}}\lambda\sigma_{g1}^* \end{pmatrix}, H\begin{pmatrix}\sigma^2 q & 0 \\ 0 & 2\lambda^2\sigma_{g1}^{*2} + \sigma_q^2\end{pmatrix}H'\right) \quad (26)$$

and $P(k_2,k_3|\varepsilon^* = \text{'no-edge'},\theta^*)$ being a normal distribution

$$N\left(H\begin{pmatrix}0\\0\end{pmatrix}, H\begin{pmatrix}\sigma^2 q & 0 \\ 0 & 2\sigma_{g1}^{*2} + \sigma_q^2\end{pmatrix}H'\right) \quad (27)$$

where

$$H = \begin{pmatrix}\cos\theta^* & \sin\theta^* \\ -\sin\theta^* & \cos\theta^*\end{pmatrix} \quad (28)$$

The conditional probability $P(A,B|\varepsilon^* = \text{'edge'},\theta^*)$ is

$$N\left(\begin{pmatrix}\mu_A^*\\0\end{pmatrix}, T\begin{pmatrix}0 & 0\\0 & \sigma_s^{2*}\end{pmatrix}T' + \begin{pmatrix}\sigma_{\eta A}^2 & 0\\0 & \sigma_{\eta B}^2\end{pmatrix}\right) \quad (29)$$

and the conditional probability $P(A,B|\varepsilon^* = \text{'no-edge'}, \theta^*)$ is

$$N\left(\begin{pmatrix}0\\0\end{pmatrix}, T\begin{pmatrix}\sigma_r^{2*} & 0\\0 & \sigma_s^{2*}\end{pmatrix}T' + \begin{pmatrix}\sigma_{\eta A}^2 & 0\\0 & \sigma_{\eta B}^2\end{pmatrix}\right) \quad (30)$$

where

$$T = \begin{pmatrix}\dfrac{\rho}{\sqrt{1+\rho^2}} & \dfrac{-1}{\sqrt{1+\rho^2}} \\ \dfrac{1}{\sqrt{1+\rho^2}} & \dfrac{\rho}{\sqrt{1+\rho^2}}\end{pmatrix}$$

and ρ is the distance along the direction θ between (r,c) and $(0,0)$ as defined in Lee [13].

The probability $P(k_2,\ldots,k_{10})$ can be expressed in terms of $P(k_2,k_3|\varepsilon^*,\theta^*)$ and $P(A,B|\varepsilon^*,\theta^*)$.

$$\begin{aligned} P(k_2,\ldots,k_{10}) &= \int_{\theta^*} P(k_2^*,k_3^*|\varepsilon^* = \text{'no-edge'},\theta^*)P(A,B|\varepsilon^*,\theta^*) \\ &\quad * P(k_4,\ldots,k_{10}|A,B) * \frac{(1-P(\text{edge}))}{2\pi}\,d\theta^* \\ &+ \int_{\theta^*} P(k_2^*,k_3^*|\varepsilon^* = \text{'edge'},\theta^*)P(A,B|\varepsilon^*,\theta^*) \\ &\quad * P(k_4,\ldots,k_{10}|A,B) * \frac{P(\text{edge})}{2\pi}\,d\theta^*. \end{aligned} \quad (31)$$

From (19), (21), (22), (23), and (24) we know that the term we used for the local edge probability is $(P(k_0|\varepsilon_0^*,\theta_0^*))/(P(\underline{k}_0))$. And, when we compute this local probability, the probability $P(k_4,\ldots,k_{10}|A,B)$ is cancelled out by dividing (25) by (31) and need not be computed. The only probability we need is the prior edge probability $P(\varepsilon^* = \text{'edge'})$ (or $P(\text{edge})$).

The edge consistency function $a(\varepsilon_0^*,\theta_0^*,\varepsilon_1^*,\theta_1^*)$ is defined separately in three different cases.

(1) $\varepsilon_0^* = $ *'edge' and* $\varepsilon_1^* = $ *'edge'*. Due to the low cumulative curvature requirement of an edge line in a small neighborhood, it is reasonable to assume that $a(\varepsilon_0^*,\theta_0^*,\varepsilon_1^*,\theta_1^*)$ has maximal value when the edge direction at the immediate adjacent neighbor agrees with that at the center, d, and the expected neighbor direction agrees with the true neighbor direction, d_1. To satisfy these requirements, we define

$$a(\varepsilon_0^* = \text{'edge'},\theta_0^*,\varepsilon_1^* = \text{'edge'},\theta_1^*) = \frac{d(\theta_0^*,\theta_1^*) + d_1(\theta_0^*, M\pi/4)}{2} \quad (32)$$

where d and d_1 are defined by

$$d(\alpha,\beta) = \frac{\cos(\alpha-\beta)+1}{2}$$

$$d_1(\alpha,\beta) = \frac{\cos 2(\alpha-\beta)+1}{2} \quad (33)$$

and M is the adjacent edge position index in the following order

5	4	3
6	0	2
7	8	1.

In $a(\varepsilon_0^*,\theta_0^*,\varepsilon_1^*,\theta_1^*)$, the center position is the position of the pixel associated with ε_1^*, θ_1^*, different positions of pixel with $\varepsilon_0^*,\theta_0^*$ correspond to different M values. The measure d is 0 (smallest value) when the difference between two angle operands is $\pm\pi$ (in opposite directions). However, when the difference between two angle operands is $\pm\pi$, the measure d_1 is 1 (highest value), since the neighbor direction difference of one of the two adjacent neighbor pixels which agree with the given edge direction is $\pm\pi$.

As α approaches β the nonlinear edge consistency function imposes less penalty (higher value) then the absolute difference function which computes absolute difference between two angles. Conversely, it gives more penalty when the angle difference is large. In this way the angle quantization effect due to the rectangular grid layout of the pixels is minimized.

(2) $\varepsilon_0^* =$ *'edge' and* $\varepsilon_1^* =$ *'no-edge'*. The edge direction consistency constrains on the 'edge' to 'no-edge' case is in general much weaker than the constraint on the 'edge' to 'edge' case. Training on a set of application specific images will in general be the best way to obtain the consistency function. However, for the purpose of completeness, we suggest a suboptimal yet general applicable function.

$$a(\varepsilon_0^* = \text{'edge'}, \theta_0^*, \varepsilon_1^* = \text{'no-edge'}, \theta_1^*) = \frac{d(\theta_0^*, \theta_1^*) + d_1(\theta_0^*, N\pi/4)}{2} \qquad (34)$$

where N is the adjacent edge position index as shown below

$$\begin{array}{ccc} 3 & 2 & 1 \\ 4 & 0 & 8 \\ 5 & 6 & 7. \end{array}$$

In $a(\varepsilon_0^*, \theta_0^*, \varepsilon_1^*, \theta_1^*)$, the center position is the position of pixel with $\varepsilon_1^*, \theta_1^*$, different positions of pixel with $\varepsilon_0^*, \theta_0^*$ correspond to different N values.

(3) $\varepsilon_0^* =$ *'no-edge' and* $\varepsilon_1^* =$ *'no-edge'*. For the purpose of completeness, we define

$$a(\varepsilon_0^* = \text{'no-edge'}, \theta_0^*, \varepsilon_1^* = \text{'no-edge'}, \theta_1^*) = \max\left\{(1 - a(\varepsilon_0^* = \text{'edge'}, \theta_0^*, \varepsilon_1^* = \text{'edge'}, \theta_1^*)), \frac{1}{2\pi}\right\}. \qquad (35)$$

If the edge directions are consistent, the function returns a low value and vice versa. We set a lower bound $(1/(2\pi))$ for the function to account for the cases of no-edge pixel pair with consistent edge direction.

2.4. *Implementation considerations*

To implement the recursive algorithm for $g_{U_{rc}}(\varepsilon_{rc}^*, \theta_{rc}^*)$, an image column is appended to the left of the image (we name it column 0) and both $g_{U_{r0}}(\varepsilon_{r0}^* = \text{'edge'}, \theta_{r0}^*)$ and $g_{U_{r0}}(\varepsilon_{r0}^* = \text{'no-edge'}, \theta_{r0}^*)$ are initialized to 1. Similarly, we append a row at the top of the image ($r = 0$) and initialize both $g_{U_{0c}}(\varepsilon_{0c}^* = \text{'edge'}, \theta_{0c}^*)$ and $g_{U_{0c}}(\varepsilon_{0c}^* = \text{'no-edge'}, \theta_{0c}^*)$ to 1. The consistency function a outputs one if its inputs include at least one appended boundary pixels. Starting from pixel position (1, 1), $g_{U_{ij}}(\varepsilon_{ij}^*, \theta_{ij}^*)$ can be recursively determined:

$$g_{U_{11}}(\varepsilon_{11}^*, \theta_{11}^*) = \frac{P(\underline{k}_{11}|\varepsilon_{11}^*, \theta_{11}^*)}{P(\underline{k}_{11})},$$

$$g_{U_{12}}(\varepsilon_{12}^*, \theta_{12}^*) = \frac{P(\underline{k}_{12}|\varepsilon_{12}^*, \theta_{12}^*)}{P(\underline{k}_{12})} * \max\left\{\sum_{\varepsilon_{11}^*, \theta_{11}^*} \frac{P(\underline{k}_{11}|\varepsilon_{11}^*, \theta_{11}^*)}{P(\underline{k}_{11})} * a(\varepsilon_{11}^*, \theta_{11}^*, \varepsilon_{12}^*, \theta_{12}^*\right.$$
$$\vdots \quad \vdots \quad \vdots \qquad (36)$$

There are two $(P(\underline{k}|\varepsilon^*, \theta^*))/P(\underline{k}))$ ratios, one for edge state = 'edge', one for edge state = 'no-edge'. Since

$$\begin{aligned} P(\underline{k}) &= \int_{\theta^*} P(\underline{k}|\varepsilon^* = \text{'edge'}, \theta^*) \\ &\quad P(\theta^*|\varepsilon^* = \text{'edge'})P(\text{edge})d\theta^* \\ &\quad + \int_{\theta^*} P(\underline{k}|\varepsilon^* = \text{'no-edge'}, \theta^*) \\ &\quad P(\theta^*|\varepsilon^* = \text{'no-edge'})P(\text{no-edge})d\theta^* \\ &= \frac{P(\text{edge})}{2\pi}\int_{\theta^*} P(\underline{k}|\varepsilon^* = \text{'edge'}, \theta^*)d\theta^* \\ &\quad + \frac{(1 - P(\text{edge}))}{2\pi}\int_{\theta^*} P(\underline{k}|\varepsilon^* = \text{'no-edge'}, \theta^*)d\theta^*. \end{aligned} \qquad (37)$$

Let θ_0^* be the θ^* which maximizes $P(\theta^*, \underline{\varepsilon}^*|k)$ (the observed gradient direction of a pixel). To simplify the computation, we define $P(\underline{k}|\varepsilon^* = \text{'edge'}, \theta^*)$ and $P(\underline{k}|\varepsilon^* = \text{'no-edge'}, \theta^*)$ such that the summation over all possible edge direction of the probability is 2π times the probability as $\theta^* = \theta_0^*$, or:

$$\int_{\theta^*} P(\underline{k}|\varepsilon^*, \theta^*)d\theta^* = 2\pi P(\underline{k}|\varepsilon^*, \theta_0^*). \qquad (38)$$

Hence

$$\begin{aligned} p(\underline{k}) &= P(\text{edge})P(\underline{k}|\varepsilon^* = \text{'edge'}, \theta_0^*) \\ &\quad + (1 - P(\text{edge}))P(\underline{k}|\varepsilon^* = \text{'no-edge'}, \theta^*). \end{aligned} \qquad (39)$$

In the case of $\varepsilon^* =$ 'edge', the local edge probability ratio is then

$$\frac{P(\underline{k}|\varepsilon^* = \text{'edge'}, \theta_0^*)}{P(\underline{k})} = \frac{P(\underline{k}|\varepsilon^* = \text{'edge'}, \theta_0^*)}{P(E)P(\underline{k}|\varepsilon^* = \text{'}E\text{'}, \theta_0^*) + (1 - P(E))P(\underline{k}|\varepsilon^* = \text{'}N\text{'}, \theta^*)} \qquad (40)$$

which is

$$\begin{cases} = 1 \text{ if } P(\underline{k}|\varepsilon^* = \text{'edge'}, \theta_0^*) = P(\underline{k}|\varepsilon^* = \text{'no-edge'}, \theta_0^*) \\ > 1 \text{ if } P(\underline{k}|\varepsilon^* = \text{'edge'}, \theta_0^*) > P(\underline{k}|\varepsilon^* = \text{'no-edge'}, \theta_0^*) \\ < 1 \text{ if } P(\underline{k}|\varepsilon^* = \text{'edge'}, \theta_0^*) < P(\underline{k}|\varepsilon^* = \text{'no-edge'}, \theta_0^*). \end{cases}$$

This gives the meaning of this local probability ratio. If the observed facet parameters of a pixel support in favor of 'edge' state, the local probability ratio will be greater than one (neutral). Otherwise, it will be less than one. When the observed facet data do not support either state, the probability ratio is one. This also explains why paths are initialized by the value one. Similar relationship holds for the $\varepsilon^* =$ 'no-edge' case.

The implementation of the recursive algorithm for $g_{L_{rc}}(\varepsilon^*_{rc}, \theta^*_{rc})$ is similar to the one we described above. In the derivations we treat context of 'edge to edge', 'edge to no-edge', and 'no-edge to no-edge' equally. However, in the context of edge lines 'edge to edge' consistency is much stronger than 'no-edge to edge' and 'no-edge to no-edge' consistency. We endeavor to remedy this unequal strength of context by emphasizing the 'edge to edge' context. We require that an edge path possesses all 'edge' pixels and on the other hand, require a no-edge path to have all 'no-edge' pixels. Moreover, the g and h functions (see equations (5) and (17)) are non-zero only when the edge direction on these functions are the angle θ^*_0 which maximizes $P(\theta^* | \underline{k})$. In order to balance local information and context information we set limitations on the context influence. We set a lower bound of influence $\epsilon_{\min}$ and an upper bound of influence $\epsilon_{\max}$, where $\epsilon_{\min} < 1 < \epsilon_{\max}$. The maximum possible value of any g and h functions are limited to $\epsilon_{\max}$ and the minimum possible value of any g and h functions are limited to $\epsilon_{\min}$.

3. GENERAL EDGE EVALUATION

3.1. *Edge coherence measure*

In order to make decisions about the performance of an edge operator, its performance score S must be estimated:

$$S = \frac{1}{|I|}\Bigg[\sum_{\varepsilon^*_{ij}, \theta^*_{ij}} S(\varepsilon^*_{ij}, \theta^*_{ij}, \varepsilon_{ij} = \text{'edge'}, \theta_{ij}: (i,j) \in A) P(\varepsilon^*_{ij}, \theta^*_{ij}: (i,j) \in A \mid E, \theta) + \sum_{\varepsilon^*_{ij}, \theta^*_{ij}} S(\varepsilon^*_{ij}, \theta^*_{ij}, \varepsilon_{ij} = \text{'no-edge'}, \theta_{ij}: (i,j) \in \bar{A}) P(\varepsilon^*_{ij}, \theta^*_{ij}: (i,j) \in \bar{A} \mid E, \theta) \Bigg] \quad (41)$$

where I is the entire image set, $A = \{(i,j) | (i,j) \in I$ and $\varepsilon_{ij} = \text{'edge'}\}$, $\bar{A} = I - A$, and E and θ are the observed edge state and edge direction of the whole image.

The score function we use here computes the score on a pixel by pixel basis. That is,

$$S(\varepsilon^*_{ij}, \theta^*_{ij}, \varepsilon_{ij}, \theta_{ij}: (i,j) \in I) = \sum_{(i,j) \in I} S(\varepsilon^*_{ij}, \theta^*_{ij}, \varepsilon_{ij}, \theta_{ij}). \quad (42)$$

Using a zero–one score function for each pixel, (41) becomes

$$\frac{1}{|I|}\Bigg[\sum_{(i,j) \in A} P(\varepsilon^*_{ij} = \text{'edge'}, \theta^*_{ij} = \theta_{ij} | E, \theta) + \sum_{(i,j) \in \bar{A}} P(\varepsilon^*_{ij} = \text{'no-edge'}, \theta^*_{ij} = \theta_{ij} | E, \theta) \Bigg]. \quad (43)$$

Similar to the definition in (2), we define

$$\tilde{g}_{Z_{ij}}(\varepsilon^*_{ij}, \theta^*_{ij}) = \max_{T \in Z_{ij}} P(\varepsilon^*_{ij}, \theta^*_{ij} | \varepsilon_{kl}, \theta_{kl}: (k, l) \in T) \quad (44)$$

where T is the best local row monotonically increasing path in the local neighborhood taking the direction θ^*_{ij} through (i,j). We can define $\tilde{g}_{U_{ij}}(\varepsilon^*_{ij}, \theta^*_{ij})$ and $\tilde{g}_{L_{ij}}(\varepsilon^*_{ij}, \theta^*_{ij})$ in a similar fashion and express $\tilde{g}_{Z_{ij}}$ in terms of $\tilde{g}_{U_{ij}}$ and $\tilde{g}_{L_{ij}}$:

$$\tilde{g}_{Z_{ij}}(\varepsilon^*_{ij} = \varepsilon_{ij}, \theta^*_{ij} = \theta_{ij}) = \frac{P(\varepsilon_{ij}, \theta_{ij})}{P(\varepsilon_{ij}, \theta_{ij} | \varepsilon^*_{ij}, \theta^*_{ij})} \tilde{g}_{U_{ij}}(\varepsilon^*_{ij} = \varepsilon_{ij}, \theta^*_{ij} = \theta_{ij}) * \tilde{g}_{L_{ij}}(\varepsilon^*_{ij} = \varepsilon_{ij}, \theta^*_{ij} = \theta_{ij}). \quad (45)$$

Just as in (16), $\tilde{g}_{U_{ij}}$ can be expressed as

$$\tilde{g}_{U_{ij}}(\varepsilon^*_{ij} = \varepsilon_{ij}, \theta^*_{ij} = \theta_{ij}) = \max_{T \in U_{ij}} \sum_{\substack{\theta^*_{kl}, \varepsilon^*_{kl} \\ (i,j) \in T^-}} \prod_{(i,j) \in T} \frac{P(\varepsilon_{kl}, \theta_{kl} | \varepsilon^*_{kl}, \theta^*_{kl})}{P(\varepsilon_{kl}, \theta_{kl})} * \prod_{(i,j) \in R(T)} C(\varepsilon^*_{ij}, \theta^*_{ij}, \varepsilon^*_{kl}, \theta^*_{kl}). \quad (46)$$

Similar expressions exist for $\tilde{g}_{L_{ij}}(\varepsilon^*_{ij} = \varepsilon_{ij},\ \theta^*_{ij} = \theta_{ij})$, $\tilde{h}_{U^*_{ij}}(\varepsilon^*_{ij} = \varepsilon_{ij},\ \theta^*_{ij} = \theta_{ij})$, and $\tilde{h}_{L^*_{ij}}(\varepsilon^*_{ij} = \varepsilon_{ij},\ \theta^*_{ij} = \theta_{ij})$.

The edge probability $P(\varepsilon_{ij}, \theta_{ij})$ can be computed as

$$P(\varepsilon_{ij}, \theta_{ij}) = \int_{\theta^*_{ij}} P(\varepsilon_{ij}, \theta_{ij} | \varepsilon^*_{ij} = \text{'edge'}, \theta^*_{ij}) P(\theta^*_{ij} | \varepsilon^*_{ij} = \text{'edge'}) P(\varepsilon^*_{ij} = \text{'edge'}) d\theta^*_{ij} + \int_{\theta^*_{ij}} P(\varepsilon_{ij}, \theta_{ij} | \varepsilon^*_{ij} = \text{'no-edge'}, \theta^*_{ij}) P(\theta^*_{ij} | \varepsilon^*_{ij} = \text{'no-edge'}) P(\varepsilon^*_{ij} = \text{'no-edge'}) d\theta^*_{ij}. \quad (47)$$

In general there is no favorite edge direction for either edge or no-edge pixels. Thus

$$P(\theta^*_{ij} | \varepsilon^*_{ij} = \text{'edge'}) = P(\theta^*_{ij} | \varepsilon^*_{ij} = \text{'no-edge'}) = \frac{1}{2\pi} \quad (48)$$

and (47) becomes

$$\frac{1}{2\pi} \int_{\theta^*_{ij}} P(\varepsilon_{ij}, \theta_{ij} | \varepsilon^*_{ij} = \text{'edge'}, \theta^*_{ij}) P(\varepsilon^*_{ij} = \text{'edge'}) d\theta^*_{ij} + \frac{1}{2\pi} \int_{\theta^*_{ij}} P(\varepsilon_{ij}, \theta_{ij} | \varepsilon^*_{ij} = \text{'no-edge'}, \theta^*_{ij}) P(\varepsilon^*_{ij} = \text{'no-edge'}) d\theta^*_{ij}. \quad (49)$$

The assumption that all the observed 'edge' pixels are true 'edge' pixels has been implicitly made in Kitchen and Rosenfeld's approach [12]. Although this assumption is not exactly true, it is in general the best choice we have for edge detector evaluation. It is not feasible to evaluate the goodness of an edge detector based on a biased prior guess. Using this realistic choice, (49) becomes

$$P(\varepsilon_{ij} = \text{'edge'}, \theta_{ij}) = \frac{1}{2\pi}\int_{\theta^*_{ij}} P(\varepsilon_{ij} = \text{'edge'}, \theta_{ij} | \varepsilon^*_{rc} = \text{'edge'}, \theta^*_{rc}) d\theta^*_{rc}. \quad (50)$$

Similar to (38), a probability function is selected to make

$$P(\varepsilon_{ij} = \text{'edge'}, \theta_{ij}) = P(\varepsilon_{ij} = \text{'edge'}, \theta_{ij} | \varepsilon^*_{rc} = \text{'}E\text{'}, \theta^*_{rc} = \theta_{ij}).$$

This implies

$$\frac{P(\varepsilon_{ij} = \text{'edge'}, \theta_{ij} | \varepsilon^*_{rc} = \text{'edge'}, \theta^*_{rc} = \theta_{ij})}{P(\varepsilon_{ij} = \text{'edge'}, \theta_{ij})} = 1.$$

Similarly, we can derive

$$\frac{P(\varepsilon_{ij} = \text{'no-edge'}, \theta_{ij} | \varepsilon^*_{rc} = \text{'no-edge'}, \theta^*_{rc} = \theta_{ij})}{P(\varepsilon_{ij} = \text{'no-edge'}, \theta_{ij})} = 1. \quad (51)$$

This concludes

$$g_{U_{ij}}(\varepsilon^*_{rc} = \varepsilon_{ij}, \theta^*_{rc} = \theta_{ij}) = \max_{T \in U_{ij}} \prod_{(i,j) \in R(T)} a(\varepsilon^*_{rc} = \varepsilon_{ij}, \theta^*_{rc} = \theta_{ij}, \varepsilon^*_{ij} = \varepsilon_{kl}, \theta^*_{ij} = \theta_{kl}). \quad (52)$$

(52) allows the estimation of the local edge coherence based on the observed edge data rather than the true edge data. Although the underlying assumptions are not mentioned in their paper, Kitchen and Rosenfeld directly used observed edge data for edge detector evaluation. Their scheme considers only local edge coherence of the observed edge pixels and ignores the observed non-edge pixels. This is the reason why they need a thinness measure to balance out the continuation measure and the number of edge pixels.

In the following derivations we consider only the local coherence of 'edge' pixels. However, it is easy to extend the derivations to include all the 'non-edge' coherence in the edge evaluation scheme. The thinness measure we used which will be described in the next section is more robust than the one that Kitchen and Rosenfeld used.

By putting (51) and (52) into (45) we get

$$\begin{aligned}\tilde{g}_{Z_{ij}}(\varepsilon^*_{rc} = \varepsilon_{ij}, \theta^*_{rc} = \theta_{ij}) &= \tilde{g}_{U_{ij}}(\varepsilon^*_{rc} = \varepsilon_{ij}, \theta^*_{rc} = \theta_{ij}) * \tilde{g}_{L_{ij}}(\varepsilon^*_{rc} = \varepsilon_{ij}, \theta^*_{rc} = \theta_{ij}) \\ &= \max_{T_1 \in U_{ij}} \prod_{(i,j) \in R(T_1)} a(\varepsilon^*_{rc} = \varepsilon_{ij}, \theta^*_{rc} = \theta_{ij}, \varepsilon^*_{ij} = \varepsilon_{kl}, \theta^*_{ij} = \theta_{kl}) \\ &\quad * \max_{T_2 \in L_{ij}} \prod_{(i,j) \in R(T_2)} a(\varepsilon^*_{rc} = \varepsilon_{ij}, \theta^*_{rc} = \theta_{ij}, \varepsilon^*_{ij} = \varepsilon_{kl}, \theta^*_{ij} = \theta_{kl}).\end{aligned} \quad (53)$$

Similar to the derivations for edge detection, by some direct substitutions and mathematical manipulations, both $\tilde{g}_{U_{ij}}(\varepsilon^*_{rc} = \varepsilon_{ij}, \theta^*_{rc} = \theta_{ij})$ and $\tilde{g}_{L_{ij}}(\varepsilon^*_{rc} = \varepsilon_{ij}, \theta^*_{rc} = \theta_{ij})$ can be obtained by a dynamic programming technique carried out in the local neighborhood around pixel (i,j). The scheme of Kitchen and Rosenfeld is just a 3×3 case of this general scheme implemented in a brute force fashion.

3.2. *Edge correctness and thinness measure*

As we mentioned, Kitchen and Rosenfeld's edge evaluator disregards the correct location of the edges. To resolve this problem, a new scheme is developed in this section. This scheme considers both the observed edge data and the original grayscale image. An edge pixel is considered as having been correctly detected if its left and right (with respect to edge direction) regions are homogeneous and possess distinct grayscale means. More explicitly, they should satisfy the following criterion:

$$\max(\sigma_1^2, \sigma_2^2) < \frac{(\mu_1 - \mu_2)^2}{g} \quad (54)$$

where (μ_1, σ_1^2) and (μ_2, σ_2^2) are the grayscale mean and variance pairs of the left and right regions; g is a selected constant reflecting the minimum signal to noise ratio of any homogeneous regions in the image.

Let T be the best row monotonically increasing path starting from an upper boundary and ending at a lower boundary of the neighborhood around the given pixel (r,c) and T_u, and T_l be the best row monotonically increasing paths ending at pixel (r,c) and starting at pixel (r,c), hence $T = T_u \cup T_l$. To construct the right and left regions of pixel (r,c), we define two vectors **u** and **l** as the vectors connecting the end points of T_u and T_l and define θ_u, θ_l as the angles between **u**, **l** and the column axis **c**. Two sets T^k_u and T^k_l are defined by:

$$T^0_u = T_u, T^0_l = T_l.$$

$$T^{k+1}_u = \begin{cases} \{(r+1,c) | \forall (r,c) \in T^k_u\} & \text{if } 0^\circ \le \theta_u \le 22.5^\circ \\ \{(r+1,c+1) | \forall (r,c) \in T^k_u\} & \text{if } 22.5^\circ < \theta_u \le 67.5^\circ \\ \{(r,c+1) | \forall (r,c) \in T^k_u\} & \text{if } 67.5^\circ < \theta_u \le 112.5^\circ \\ \{(r-1,c+1) | \forall (r,c) \in T^k_u\} & \text{if } 112.5^\circ < \theta_u \le 157.5^\circ \\ \{(r-1,c) | \forall (r,c) \in T^k_u\} & \text{if } 157.5^\circ < \theta_u \le 180^\circ \end{cases} \quad (55)$$

$$T^{k+1}_l = \begin{cases} \{(r-1,c) | \forall (r,c) \in T^k_l\} & \text{if } 0^\circ \ge \theta_l \ge -22.5^\circ \\ \{(r-1,c+1) | \forall (r,c) \in T^k_l\} & \text{if } -22.5^\circ > \theta_l \ge -67.5^\circ \\ \{(r,c+1) | \forall (r,c) \in T^k_l\} & \text{if } -67.5^\circ > \theta_l \ge -112.5^\circ \\ \{(r+1,c+1) | \forall (r,c) \in T^k_l\} & \text{if } -112.5^\circ > \theta_l \ge -157.5^\circ \\ \{(r+1,c) | \forall (r,c) \in T^k_l\} & \text{if } -157.5^\circ > \theta_l \ge -180^\circ \end{cases} \quad (56)$$

where $k \in$ integer set N. The set T^k is defined as

$T_u^k \cup T_l^k$. We can also define T_u^{k-1} and T_l^{k-1} in a similar fashion. For a selected width d which is smaller than or equal to the minimum width of any meaningful region of the image, the pixels belong to the set $\cup_{i=1}^{d} T^i$ and $\cup_{i=-1}^{-d} T^i$ construct the right and the left regions. The location accuracy measure la of an observed edge point is defined as

$$la = \min\left\{1, \frac{(\mu_1 - \mu_2)^2}{g * \max(\sigma_1^2, \sigma_2^2)}\right\}. \tag{57}$$

We now introduce the edge thinness measure T. Let N_l be the set of the left adjacent 'edge' pixels of T and N_r be the set of right adjacent 'edge' pixels of T:

$$N_l = \{(i,j) | (i,j) \in T^{-1} \text{ and } \varepsilon_{ij} = \text{'edge'}\}$$
$$N_r = \{(i,j) | (i,j) \in T^{1} \text{ and } \varepsilon_{ij} = \text{'edge'}\}.$$

Let $|S|$ represent the number of elements belong to a set S. A generalized thinness measure T is defined by:

$$T = 1 - \frac{|N_l| + |N_r|}{|T^1| + |T^{-1}|} \tag{58}$$

or

$$T = 1 - \frac{2 * \min\{|N_l| + |N_r|\}}{|T^1| + |T^{-1}|} \tag{59}$$

depends on whether we allow the existence of ideal step edges (two pixel width (use (59)) or not (use(58)). Figure 2 shows the sets N_l, N_r, T^1, and T^{-1}.

The overall edge score based on local edge coherence (not including edge position correctness) is a linear combination of the continuation measure C, obtained from equation (41), and the thinness measure T.

$$E = w(T)C + (1 - w(T))T.$$

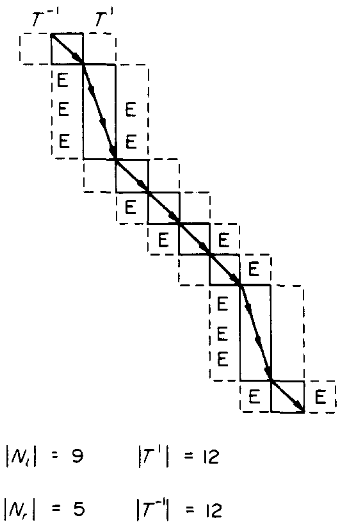

Fig. 2. This shows the sets N_l, N_r, T^1, and T^{-1}.

The w we use here is a function of T:

$$w(T) = \begin{cases} 0.2 & \text{if } 1 \geq T \geq 0.85 \\ 0.25 & \text{if } 0.85 > T \geq 0.75 \\ 0.3 & \text{if } 0.75 > T \geq 0.65. \end{cases} \tag{60}$$

We do not allow any edge to be thicker than $T < 0.65$. When T is small, the image has quite a lot of redundant edges and the edge evaluation scheme is not reliable for this kind of image and the only comment we can make is probably "they are bad edge images". In Kitchen and Rosenfeld's experiment for edge detection they did not set the lower limit on T. This results in cases of lower edge scores for higher SNR images.

4. EXPERIMENTAL RESULTS

4.1. *General edge evaluator*

We present some experiments to verify the performance of the general edge evaluator. To permit a comparison, we use similar edge detection schemes and the same noise model and test images as those of Kitchen and Rosenfeld. However we extended the edge evaluation kernel size from 3 × 3 to 5 × 5.

Three well known edge detectors are used in the experiments, the Kirsch operator [Kirsch 11], the 3 × 3 Sobel operator [Duda and Hart 3], and Neviatia's compass operator [Neviatia and Babu 17]. Two test images were used. The first one is a "vertical edge" image of size 64 × 64 pixels. It consists of a left panel with grey level 115, a right panel with grey level 140, and a single central column of intermediate grey level 128. The second one is a "ring" image consisting of concentric light rings (grey level 140) on a dark background (grey level 115). This image was originally generated as a 512 by 512 image with a central dark circle of radius 64, surrounded by three bright rings of width 32, these being separated by two dark rings of the same width, with a dark surround. The decision as to whether a pixel should be light or dark is based on its Euclidean distance from the center of the image. This image was then reduced to size 128 by 128, by replacing each 4 by 4 block with a single pixel having the average grey level of the block.

Independent zero-mean Gaussian noise was added to each of the test images at seven different signal to noise ratios: 1, 2, 5, 10, 20, 50, and 100. The signal-to-noise ratio (SNR) is defined by SNR = $(h/\sigma)^2$, where h is the edge contrast (in this case 25) and σ is the standard deviation of the noise. Figures 3 and 4 show these two test images and their noisy images with different SNR.

In the experiment, we applied each operator to the test images at the seven different signal-to-noise ratios, and at each noise level the threshold was adjusted to maximize the edge evaluation score E. The local neighborhood size for edge evaluation is selected as 5 × 5. The size is larger than Kitchen and Rosenfeld's scheme which can only deal with 3 × 3 neighborhood. Figures 5 and 6 show the evaluation results for

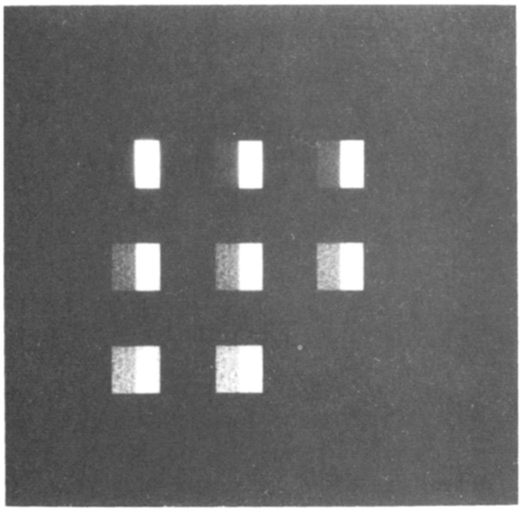
Fig. 3. Vertical edge test image, with various levels of noise. From left to right, top to bottom: no noise, SNR 100, 50, 20, 10, 5, 2, and 1.

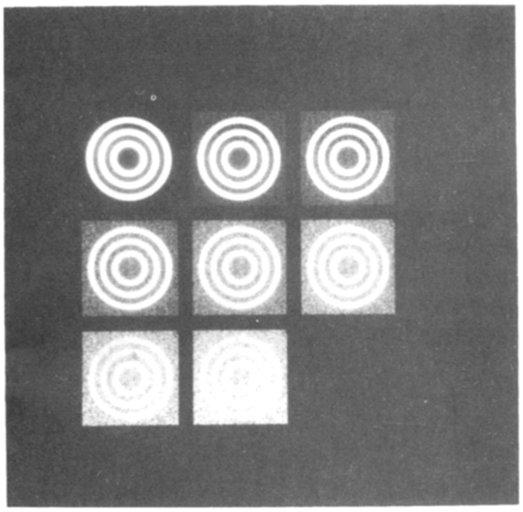
Fig. 4. Rings test image, with various levels of noise. From left to right, top to bottom: no noise, SNR 100, 50, 20, 10, 5, 2, and 1.

different edge operators.

The results based on the local edge coherence measure give compass edge operator high score over the other two operators when the noise level is low to medium. However, as noise goes high the compass operator becomes the worst one. This is due to the fact that the Nevitia's compass edge detection scheme performs not only edge detection but also edge line linking. When the image is not too noisy the edge linking process can improve the edge image. Whereas the image is pretty noisy, the edge linking process tends to link wrong edge lines. The Kirsch operator performs better than the Sobel operator when the SNR is higher than 10. It performs worse than the Sobel operator when the SNR is greater than 10. It is noted that the performance score curves of these operators for the low SNR images are not as flat as the curves of Kitchen and Rosenfeld's evaluator (see Fig. 5). Because of the lower bound we used for the thinness measure *T*. The edge images of these operators on the vertical edge image are shown in Figs 7(a), 8(a) and 9(a), respectively. It is noted that a visual evaluation of these edge images is consistent with that which we obtained quantitively based on the evaluator.

The result of the edge position correctness measure for different operators are shown in Fig. 10. The figure shows that for this particular image the edge correctness measure is consistent with the edge coherence measure. Any combination of them will yield almost equivalent evaluation score of each operator.

The evaluation scores based on the local edge coherence measure on the ring test images are different from those of the vertical edge images for the operators. The Sobel operator has uniformly best scores, while Nevatia's compass operator performs the worst. The low score of the compass operator is partially caused by the single width edge based edge linking process and is even further degraded by the non-equivalent edge directions of adjacent edge pixels. Since we set the lower bound on the thinness measure *T*, the performance score curves of these operators on the low SNR images are not as flat as the performance curves of Kitchen and Rosenfeld's evaluator. The edge images after applying these operators to the rings image are shown in Figs 7(b), 8(b) and 9(b), respectively.

The results of the edge position correctness measure for different operators on the rings images are shown in Fig. 11. It shows that if we were to combine the edge correctness measure with the edge coherence measure, we will increase the score of the compass operator because its performance is not really as bad as it appears in the edge coherence score. Thus, a combination of the two measures provides more reliable edge score.

4.2. *Context dependent edge detector*

To understand the performance of the context dependent edge detector, we examine the behavior of the context edge detector on one well structured simulated image and two real images. We then compare the results with the context free second derivative zero-crossing edge operator [Haralick, 7] to see how and in what degree the context information can improve the operator.

The simulated test image is a noisy bar image. The image size is 100×50 pixels. The pixel intensity is 0 for dark bars and 175 for white bars. A 2×2 averaging is applied to this image to simulate ideal single pixel width edge lines. A zero mean Gaussian noise with standard deviation 40 is then added to this image. We fit each 5×5 neighborhood of the test image by a cubic polynominal and then apply the context dependent operator and second derivative zero crossing edge operator to it. In order to quantitively see

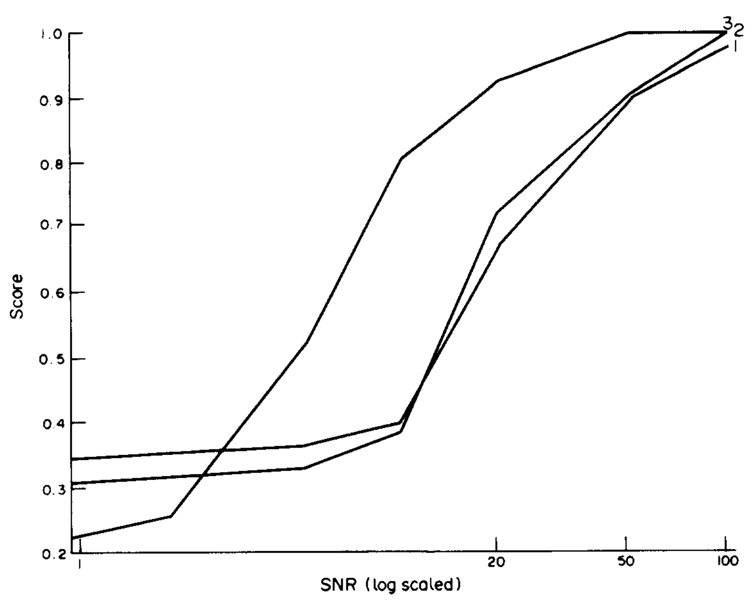

Fig. 5. The edge coherence score for different edge operators. The test image is the vertical edge image, where 1 is the score for Sobel operator, 2 is the score for Kirsch operator, and 3 is the score for Nevitia's compass operator.

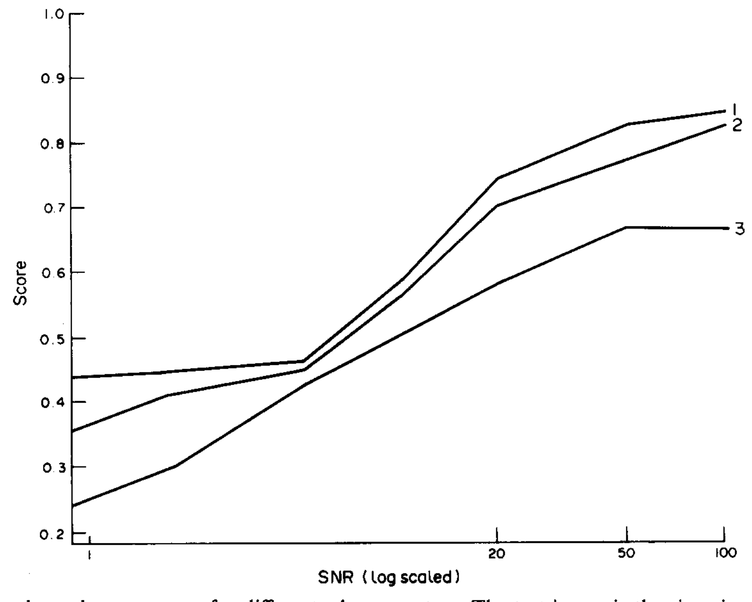

Fig. 6. The edge coherence score for different edge operators. The test image is the rings image, where 1 is the score for Sobel operator, 2 is the score for Kirsch operator, and 3 is the score for Nevitia's compass operator.

the difference in performance of these two schemes, we measure for the whole image the conditional probabilities $P(E'|E^*)$, $P(E^*|E')$, $P(E'|\bar{E}^*)$, and $P(\bar{E}'|E^*)$ where E' and $\bar{E}'$ designate the assigned 'edge' and 'no-edge' pixel sets and E^* and $\bar{E}^*$ designate the true 'edge' and 'no-edge' pixel sets. When determining $P(E'|E^*)$ and $P(E^*|E')$, the adjustable parameters of each edge operator are chosen to equalize these two conditional probabilities $(P(E'|E^*) \approx P(E^*|E'))$. The performance in terms of $P(E'|E^*)$ and $P(E^*|E')$ are shown in Table 1. When determining $P(E'|\bar{E}^*)$ and $P(\bar{E}'E^*)$, the adjustable parameters of each edge operator are chosen to equalize these two conditional probabilities $(P(E'|\bar{E}^*) \approx P(\bar{E}'|E^*))$. The probability $(P(E'|\bar{E}^*)$ for the context free operator is 0.02 while the probability for the context operator is 0. The results show that the context scheme performs much better than the context free operator.

We now apply the context dependent edge detector to two 3-D range images (128 × 128 pixels) of man-made objects. To these images a zero mean Gaussian noise of standard deviation 30 is added. The noisy

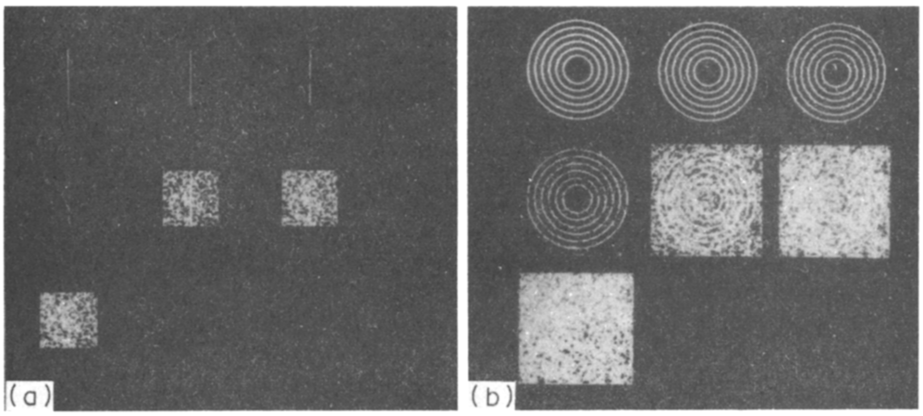
Fig. 7. Edge results from Sobel operator for (a) vertical edge test image and (b) rings test image with various levels of noise. From left to right, top to bottom: no noise, SNR 100, 50, 20, 10, 5, 2, and 1.

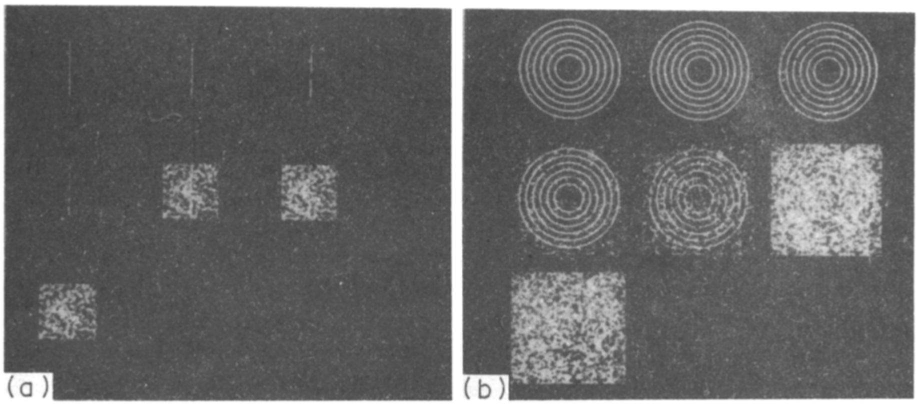
Fig. 8. Edge results from Kirsch operator for (a) vertical edge test image and (b) rings test image with various levels of noise. From left to right, top to bottom: no noise, SNR 100, 50, 20, 10, 5, 2, and 1.

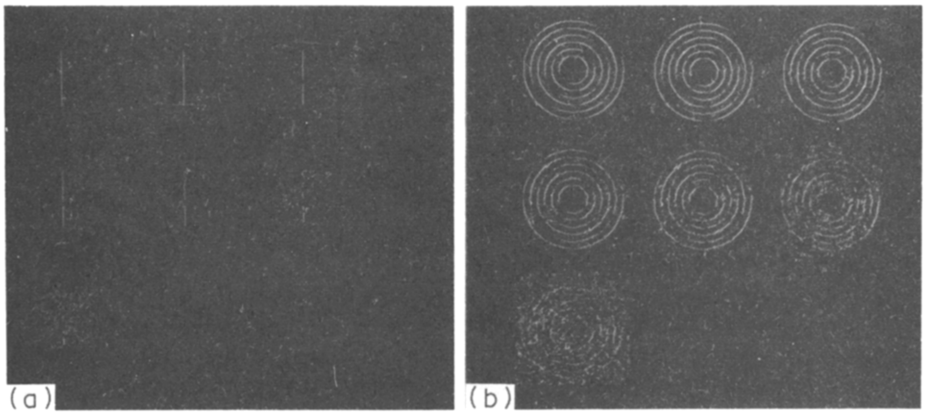
Fig. 9. Edge results from Nevitia's compass operator for (a) vertical edge test image and (b) rings test image with various levels of noise. From left to right, top to bottom: no noise, SNR 100, 50, 20, 10, 5, 2, and 1.

Table 1. $P(E'/E^*)$ and $P(E^*/E')$ values of the context dependent edge operator and the context free edge operator

Operator\probability	$P(E' \mid E^*)$	$P(E^* \mid E')$
Context dependent	0.8600	0.8600
Context free	0.6950	0.6814

images are shown in Figs 12(a), and 13(a). A 5 × 5 cubic polynominal fitting is applied to the noisy images followed by the applications of both the context free second derivative zero-crossing edge operator and the context dependent edge operator. The context bounds we used for both images are

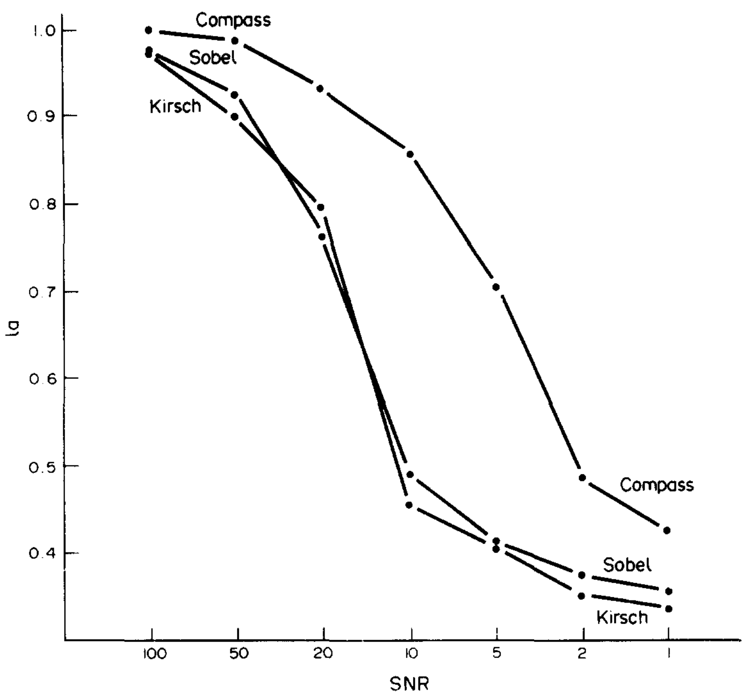

Fig. 10. The edge location accuracy measures on the vertical edge image for different edge operators.

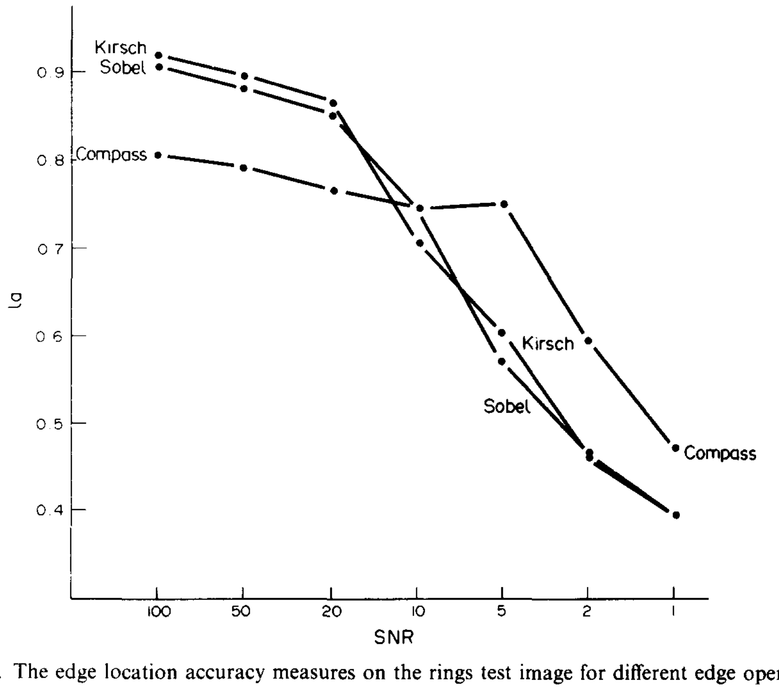

Fig. 11. The edge location accuracy measures on the rings test image for different edge operators.

$\epsilon_{min} = 0.1$ and $\epsilon_{max} = 10$. The results of the context free edge operator are shown in Figs 12(b), and 13(b). The results of the context dependent edge operator are shown in Figs 12(c), and 13(c). It can be easily verified visually that the edge images of the context dependent edge operator show better connectivity and much less noise than the context free edge operator.

In order to see the performance of the context dependent edge operator under different noise levels, the context dependent and the context free edge operators are applied to images with noises of standard deviations 10, 20, 30, 40, and 50, respectively. The general edge evaluator is employed to measure the performance scores of both edge operators. The edge results which maximize the edge score are shown in Fig. 14. It is also easy to verify that the context edge operator has better performance over the context

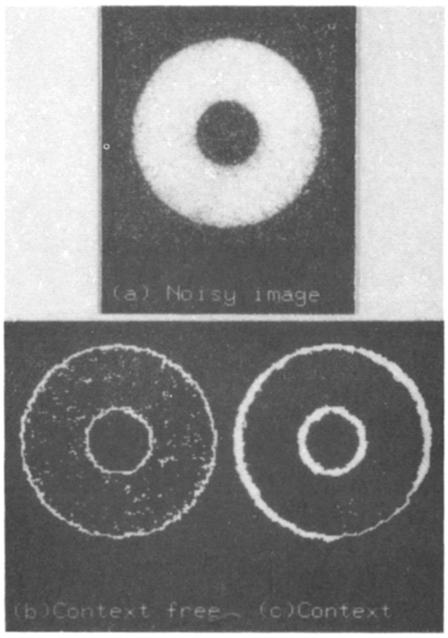

Fig. 12. The noisy object image 1(a), its context free edge image (b), and context dependent edge image (c). The window size for cubic polynomial fitting is 5 × 5.

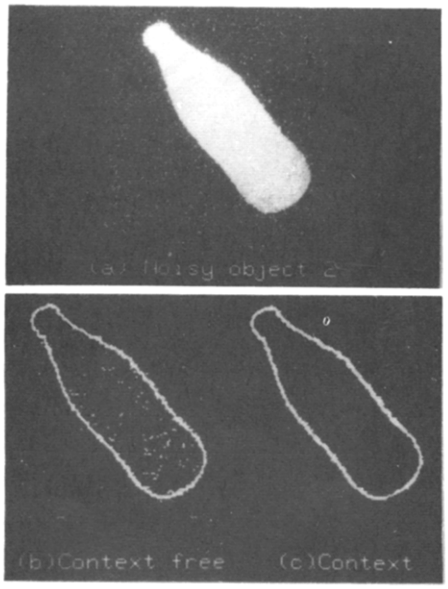

Fig. 13. The noisy object image 2(a), its context free edge image (b), and context dependent edge image (c). The window size for cubic polynomial fitting is 5 × 5.

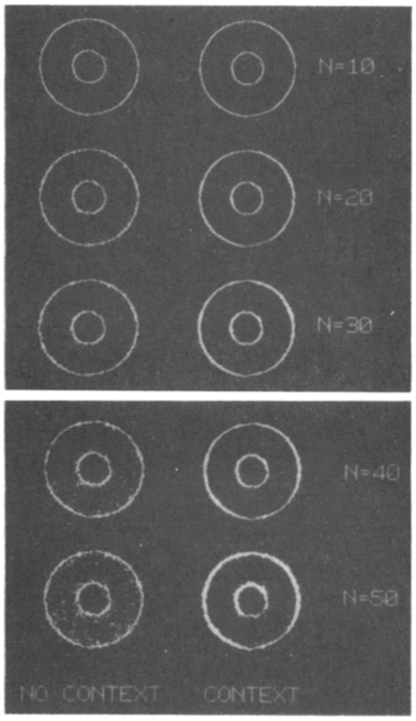

Fig. 14. Shows the edge images of the context dependent edge operator and the context free edge operator applied to the image shown in Fig. 12 with noise standard deviations 10, 20, 30, 40, and 50, respectively.

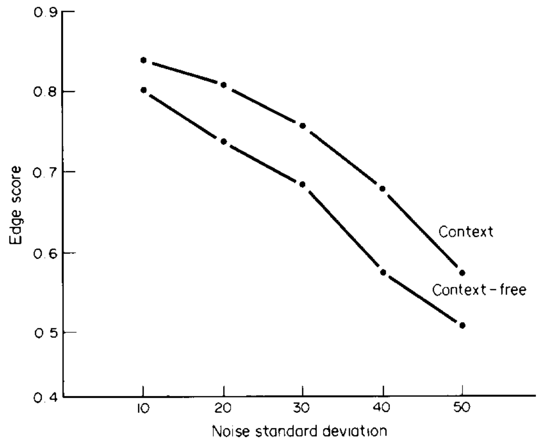

Fig. 15. Shows the edge scores of the context dependent edge operator and the context free edge operator applied to the image shown in Fig. 12 with noise standard deviations 10, 20, 30, 40, and 50, respectively.

free edge operator by a visual evaluation. The edge score curves are shown in Fig. 15.

The edge consistency function we used is a linear combination of the functions $d(\alpha, \beta)$ and $d_1(\alpha, \beta)$ which are defined by (33). In order to see the performance of the context edge operator when employing different consistency functions, we use d' and d_1' to replace d and d_1 for the edge consistency function. d' and d_1' are defined as power of d and d_1, respectively. Thus,

$$d'(\alpha, \beta) = (d(\alpha, \beta))^{\text{power}}$$

and

$$d_1'(\alpha, \beta) = (d_1(\alpha, \beta))^{\text{power}}.$$

We apply the context operator to the image shown in Fig. 12(a) by employing consistency functions of powers 0.3, 0.5, 1.0, 1.5, 2.0, 3.0, 4.0, and 5.0, respectively All the parameter settings except the power of the consistency function are kept at the same values as we used to produce Fig. 12(c). The edge results are shown in Fig. 16.

It is found that when the power is small, the detected edge lines appear very thick and the connectivity of the edge lines is good. Conversely, as the power of the consistency function increases, the detected edges are thin and broken into pieces. The best edge image among them is the image of power one. It is not difficult to explain this phenomenon. When the power of the consistency function is small the consistency function outputs high values in a wide range of angle differences. Thus, the context information is relatively strong in the edge detection process and thus links pixels nearby edge pixels. On the other hand, when the power of the consistency function is large, the context information is relatively weak in the edge detection process and can not fill the gaps in the middle of the edge lines.

Finally, we apply the context dependent edge operator to an image corrupted by a signal dependent noise. The signal dependent noise image is generated by the function

$$SDN(f(r,c)) = N\left(f(r,c), \frac{f(r,c)}{5} + 10\right),$$

where $N(a,b)$ returns a normal random number with mean a and standard deviation b. We apply the signal dependent noise to object image 1. The noisy image is shown in Fig. 17(a). We fit each 5×5 neighborhood of the noisy image by a bivariate cubic polynomial and apply both the context-free and the context dependent operators to the fitted image. The edge results are shown in Figs 17(b) and (c). The edge image produced by the context dependent operator appears better than the edge image produced by the context-free operator.

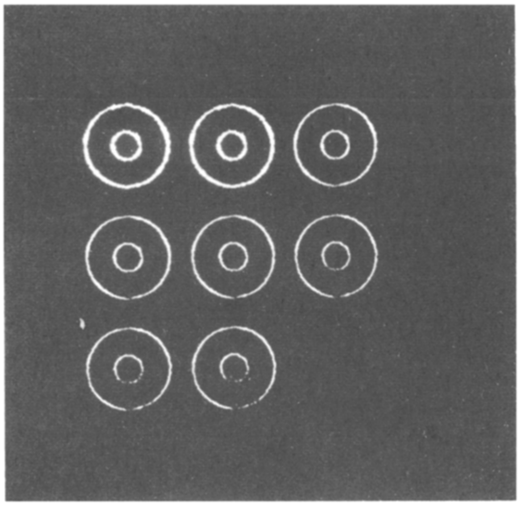
Fig. 16. The context operator applies to the noisy object image 1 with different powers of the consistent function. The powers are from left to right, top to bottom: 0.3, 0.5, 1.0, 1.5, 2.0, 3.0, 4.0, and 5.0.

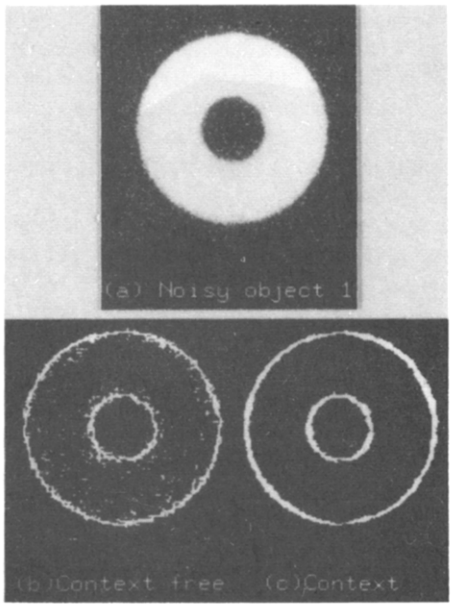

Fig. 17. The object image 1 corrupted by a signal dependent noise (a), its context free edge image (b), and context dependent edge image (c). The window size for cubic polynomial fitting is 5 × 5.

5. CONCLUSIONS

We have developed an edge detection scheme from a Bayesian theoretic framework. The edge detection makes use of the edge context of the entire image. For a given pixel the edge context of the whole image related to this pixel is organized as monotonically increasing paths which begin at any border pixels of the image above the selected pixel, pass through the selected pixel, and end at some border pixels of the image below the selected pixel. We will assign a pixel edge state 'edge' if the edge probability of the best 'edge' path is higher than the average probability of the best 'no-edge' paths.

We derived the algorithms of finding the best paths as a recursive scheme. It starts with a top down left–right scan of the image followed by a right–left scan. It then performs a bottom up right–left scan of the image followed by a left–right scan.

Based on the same framework, a general method for evaluating the quality of edge detector output is developed. Three measures are presented. The continuous measure evaluates how good the edge is formed as continuous lines. The thinness measure evaluates how thin the edge lines are. The location accuracy measure evaluates the position correctness of the edge lines. A robust scheme is introduced to combine the continuous and thinness measures.

The general edge evaluator does not require knowledge of the true location of edges and it can make the evaluation based on local edge coherence in a larger than 3×3 neighborhood. It can be part of a feedback mechanism which allows automatic adjustment of the parameters, such as thresholds, for optimum detection of edges in real images.

Experiments were performed to illustrate the validity of the context edge detector and the general edge evaluator. We have compared the performance of the context edge detector with the context free second directional derivative zero-crossing edge operator. The results show that the context edge detector has superior performance.

The approach of using full context and local coherence can be extended to the detection and evaluation of other local image features.

REFERENCES

1. Abdou, I. E.; Pratt, W. K. Qualitative design and evaluation of enhancement/thresholding edge detector. *Proc. IEEE* **67(5):** 753–763; 1979.
2. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Analysis Mach. Intell.* **PAMI-8** (6): 679–698; 1986.
3. Duda, R. O.; Hart, P. E. *Pattern Classification and Scene Analysis.* New York: Wiley; 1973.
4. Fram, J. R.; Deutsch, E. S. On the quantitative evaluation of edge detection schemes and their comparisons with human performance. *IEEE Trans. Comput.* **C-24(6):** 616–627; 1975.
5. Haralick, R. The digital edge. *Proc. IEEE* 1981 *Conf. Pattern Recognition Image Processing.* New York: IEEE Computer Society; 285–294; 1981.
6. Haralick, R. Zero-crossing of second directional derivative edge operator. *Soc. Photogrammetric Instrument. Engr Symp. Robot Vision*; Washington D.C.; 1982.
7. Haralick, R. Digital step edges from zero crossing of second directional derivatives. *IEEE Trans. Pattern Analysis Mach. Intell.* **PAMI-6(1):** 58–68; 1984.
8. Haralick, R. Decision making in context, *IEEE Trans. Pattern Analysis Mach. Intell.* **PAMI-5(4):** 417–428; 1983.
9. Haralick, R.; Watson, L. T. A facet model for image data. *Comput. Graphics Image Process.* **15:** 113–129; 1981.
10. Haralick, R.; Lee, J. S. J. Neighborhood context dependent edge detection. *IEEE Trans. Pattern Analysis Mach. Intell.* (to appear).
11. Kirsch, R. Computer determination of the constituent structure of biological images. *Comput. Biomed. Res.* 4: 315–328; 1971.
12. Kitchen, L.; Rosenfeld, A. Edge evaluation using local edge coherence. *IEEE Trans. Syst. Man Cybernetics* **SMC-11** (9): 597–605; 1981.
13. Lee, J. S. J. Edges From Image. VPI & SU; PH.D. Dissertation; 1985.
14. Marr, D.; Hildreth, E. Theory of edge detection. *Proc. R. Soc. Lond. B* **207:** 187–217; 1980.
15. Martelli, A. An application of heuristic search methods to edge and contour detection. *Communs ACM* **19:** 73–83; 1976.
16. U. Montanari, On the optimal detection of curves in noisy pictures. *Communs ACM* **14:** 335–345; 1971.
17. Nevatia, R.; Babu, K. R. Linear feature extraction and description. *Comput. Graphics Image Process.* 257–269; 1980.
18. Peli, T.; Malah, D. A study of edge detection algorithms. *Comput. Graphics Image Processing* **20:** 1–21; 1982.
19. Zucker, S. W.; Hummel, R. A.; Rosenfeld, A. An application of relaxation labeling to line and curve enhancement. *IEEE Trans. Comput.* **C-26**(4): 394–403; 1977.

About the Author—ROBERT M. HARALICK was born in Brooklyn, New York on 30 September 1943. He received a B.A. degree in Mathematics from the University of Kansas in 1964, a B.S. degree in Electrical Engineering in 1966 and a M.S. degree in Electrical Engineering in 1967. In 1969, after completing his Ph.D. at the University of Kansas, he joined the faculty of the Electrical Engineering Department there where he last served as Professor from 1975 to 1978. In 1979 Dr Haralick joined the Electrical Engineering Department at Virginia Polytechnic Institute and State University where he was a Professor and Director of the Spatial Data Analysis Laboratory. From 1984 to 1986 Dr Haralick served as Vice President of Research at Machine Vision International, Ann Arbor, MI. Dr Haralick now occupies the Boeing Clairmont Egtvedt Professorship in the Department of Electrical Engineering at the University of Washington.

Professor Haralick has made a series of contributions in computer vision. In the high-level vision area, he has identified a variety of vision problems which are special cases of the consistent labeling problem. His papers on consistent labeling, arrangements, relation homomorphism, matching, and tree search translate some specific computer vision problems to the more general combinatorial consistent labeling problem and then discuss the theory of the look-ahead operators that speed up the tree search. This gives a framework for the control structure required in high-level vision problems. Recently, he has extended the forward-checking tree search technique to propositional logic.

In the low-level and mid-level areas, Professor Haralick has worked in image texture analysis using spatial gray tone co-occurrence texture features. These features have been used with success on biological cell images, X-ray images, satellite images, aerial images and many other kinds of images taken at small and large scales. In the feature detection area, Professor Haralick has developed the facet model for image processing. The facet model states that many low-level image processing operations can be interpreted relative to what the processing does to the estimated underlying gray tone intensity surface of which the given image is a sampled noisy version. The facet papers develop techniques for edge detection, line detection, noise removal, peak and pit detection, as well as a variety of other topographic gray tone surface features.

Professor Haralick's recent work is in shape analysis and extraction using the techniques of mathematical morphology. He has developed the morphological sampling theorem which establishes a sound shape/size basis for the focus of attention mechanisms which can process image data in a multiresolution mode, thereby making some of the image feature extraction processes execute more efficiently.

Professor Haralick is a Fellow of IEEE for his contributions in computer vision and image processing. He serves on the Editorial Board of *IEEE Transactions on Pattern Analysis and Machine Intelligence.* He is the computer vision area editor for *Communications of the ACM.* He also serves as associate editor for *Computer Vision, Graphics, and Image Processing* and *Pattern Recognition.*

About the Author—James S. J. Lee was born in Chang-Hwa, Taiwan in 1955. He received the B.S. degree in control engineering from National Chiao Tung University in 1977, the M.S. degree in electrical engineering from National Taiwan University in 1979, and the Ph.D. degree from Virginia Polytechnic Institute and State University, Blacksburg, VA in 1985.

From 1980 to 1981, he was an engineering officer of the Electrical Division in the R.O.C. Navy Third shipbuilding yard for the required ROTC military service. He was awarded the outstanding invention prize by the R.O.C. Navy general headquarters and the "First Prize of Invention" by the R.O.C. Ministry of Education. From 1981 to 1982 he was an Instructor in the Department of Electronic Technology. National Taiwan Institute of Technology. His activities there included teaching and robotics research. From 1982 to 1984, he was a Research Assistant at the Spatial Data Analysis Laboratory of Virginia Tech engaged in image processing and artificial intelligence research. From 1984 to 1986 he was a Research Scientist at the Research and Development Department of Machine Vision International where he conducted research in real-time machine vision algorithm and architecture. In the fall of 1986 he joined the High Technology Center of the Boeing Electronics Company as a Senior Specialist Engineer in the Information Processing Laboratory. His current research interests include adaptive image processing, real-time machine vision architecture, intelligent vision systems, distributed problem solving, machine learning, and sensor fusion.

Dr Lee is a member of Phi Kappa Phi.