Pergamon

Pattern Recognition, Vol. 31, No. 5, pp. 587–596, 1998

0031–3203/98 $19.00 + .00

PII: S0031-3203(97)00090-3

KNOWLEDGE-BASED ASSISTANT FOR THE SELECTION OF EDGE DETECTORS

D. ZIOU*,† and A. KOUKAM‡

*Département de mathématiques et d'informatique, Faculté des sciences, Université de Sherbrooke, Sherbrooke, Québec, Canada, J1K 2R1
‡Département de Génie Informatique, Institut Polytechnique de Sevenans, 90010 Belfort Cedex, France

(*Received* 6 *August* 1996; *accepted* 23 *July* 1997)

Abstract—This paper summarizes a system, called SED "Selection of Edge Detectors", which is able to automatically select edge detectors and their scales to extract a given edge. The basic organization of the SED system is a collection of edge detectors within a knowledge structure made up of the characteristics of the given edge, the properties of the detectors and the mutual relation between them. Combination of this information in the selection process provides a basis for avoiding a combinatorial search for appropriate edge detectors. The version of the system described here has been fully implemented and is now being enriched to integrate a maximum of edge detector algorithms. We illustrate our approach by an example. © 1998 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved

Edge detection   Selection of edge detectors   Rule-based system

1. INTRODUCTION

In computer vision, edge detection is the process which attempts to capture the significant properties of objects represented in the image. These properties include the discontinuities of the photometrical, geometrical and physical characteristics of objects. Such information gives rise to variations in the grey level image; the most commonly used variations are the discontinuities, called step edges, and the local extrema, called line edges.

The purpose of edge detection is the localization of these variations and the identification of the physical phenomena which gave rise to them. Edge detection must be efficient and reliable because the validity, efficiency and even the possibility of realizing subsequent processing stages are related. However, it is difficult to derive a general algorithm which is optimal according to these requirements, because the image is a discrete and noisy function. Consequently, a variety of edge detectors have been devised in the history of digital image processing, which differ both in their purpose (i.e. photometrical and geometrical properties of the edge) and their mathematical and algorithmic properties. In order to facilitate their use, various software packages have been developed.(1) In computer vision, however, we are often confronted with the problem of selecting an appropriate edge detector. Usually, the approach used consists of arbitrarily choosing a detector and using it to find all the edges in the images being processed in the target application. The scale is often fixed by trial-and-error experiments and reused for all images. It is clear that this approach does not lead to correct results. In fact, one detector running at one scale does not yield all edges of the image for the following two reasons: (1) in an image, edges are produced by different physical phenomena and do not represent the same scene information. The different proposed edge detectors have different properties and yield different results. (2) All edges cannot be correctly detected at the same scale. But, an edge detector can be run only at one scale. Thus, it is appropriate to focus on edges mathematical properties and goals by using several detectors that differ in their scales. The underlying problems are multiple; various types of knowledge and know–how about image formation and processing techniques are required to achieve an effective approach. Matsuyama(1) has discussed the six problems frequently encountered in designing image analysis systems.

Hasegawa, Kubota and Toriwaki(2) implemented an intermediate solution in the IMPRESS system that is able to choose the appropriate detector to find a given edge. It involves applying all the detectors and retaining the one which generates the most similar result to the reference edge. We propose another intermediate solution implemented in a system called Selection of Edge Detectors (SED) which is able to automatically select edge detectors and their scales to extract a given edge. There are two significant differences between the IMPRESS system and the SED system: (1) In SED, we use selection criteria which combine several sources of information (i.e. edge characteristics, detector properties, the mutual influence between edges and detectors, and the results of the detectors run) to avoid a combinatorial approach.

†Author to whom correspondence should be addressed.

(2) We consider that the edge detector is made up of smoothing and differentiation operations. In the IMPRESS system, an edge detector includes smoothing, differentiation, thresholding, and linking.

In the next section, we present an outline of our approach. Section 3 is devoted to the analysis of image structure. We give the selection criteria and the analysis of the results in Sections 4 and 5. Finally, we present a thorough study of an example.

2. SYSTEM OVERVIEW

The underlying ideas of this work are drawn from research on computer vision and the knowledge-based approach. From computer vision research, we gained experience in image analysis, specifically the design and use of edge detectors. From knowledge-based approach, we acquired suitable structures to represent knowledge and the role of edge detectors in image processing. The knowledge was acquired through study of the literature on edge detection[3–15] and also from our prior research concerning this problem.[16–21]

The definition of the selection approach and the structuring of the underlying system depend on the type of knowledge which can be used. Here, we use three sources of knowledge for the selection of detectors and their scales. The first is the properties of the edges produced by the detector and analysis of the successes and failures of this detector. The second source concerns *a priori* knowledge about the detector such as its mathematical and algorithmic properties, including linearity, invariance to rotation, duration of the impulse response of the filter, order of the differentiation operator, computational complexity, and goal. The last source is the edge characteristics such as position, form, and type. Consideration of the first source alone leads to a combinatorial solution. For example, one possible scheme is to run all available detectors at several scales and to select the best detector by comparing their results. This scheme is used in the IMPRESS system mentioned above to select the best detector. The second and third sources are interdependent and must be used together. Thus, selection is based on the mutual influence of edge characteristics and detector properties. In other words, given the edge characteristics and the detector properties, we define a set of rules for the selection of detectors and their scales. However, usually these rules are imprecise and consequently selection errors are introduced. As explained below we use the three sources of knowledge mentioned above together to minimize the possibility of selection errors.

The SED system has, as input, an edge represented by its location, the image that contains it (see Fig. 5), and the quality of the required results expressed as a set of constraints related to delocalization error and computation time. The results of this system are the characteristics of the given edge, and the detectors that are able to find it, with their scales. Figure 1 shows the main components in the SED system and their interactions. To avoid a combinatorial approach for the selection of the appropriate detectors, the SED system includes a library containing a set of tools related to edge detection (i.e. edge detectors and cleaning algorithms) and the knowledge base required for

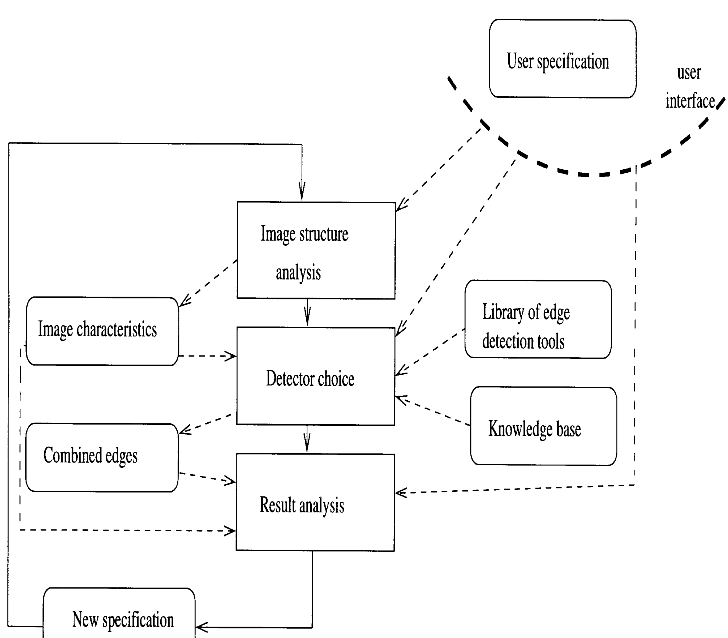

Fig. 1. Procedural control structure of the SED system.

the selection process. The first step consists of computation of the pertinent characteristics of the current reference edge and its segmentation into a set of edgels. An edgel is a set of points having homogeneous characteristics. The second step involves choosing the best detector and its parameters. This is based on its performance, the results of previously run detectors, and a set of rules which specify the influence of the attributes of an edge and the properties of the detector on its performance. In this step, the basic control structure is that of a production system in which knowledge about the selection is formulated into condition–action rules. A system process matches rules against the data computed by the image structure analysis process and when a match occurs, the best rule is fired. This triggers an action that involves a run of the selected detector. Finally, in the third step the results of the selected detector are combined with the results obtained previously. If the initial reference edge is not fully found, then the missed edgels are identified and they form the new reference edge. The parameters used in the image structure analysis are updated and the feedback route is tried once (go to the image structure analysis step). In what follows, we will give full details about the three steps of the system.

3. IMAGE STRUCTURE ANALYSIS

This step involves analysis of the input data and segmentation of the current reference edge into a set of edgels. It includes measurement of the geometrical and photometrical characteristics of the edge which influence the performance of an edge detector. The geometrical characteristics are deduced from the initial reference edge and include position, orientation, and smoothness. The photometrical characteristics are an accurate, detailed description of the variation of image intensity in the vicinity of the edge. Their definition is related to the profile of the edge. We consider two kinds of edge profile: the step edge (a discontinuity in the image intensity) and the double edge (two close discontinuities, i.e. pulse and staircase edges). The motivation behind this distinction is the fact that localization of the double edge requires more attention because the smoothing of each step edge is greatly influenced by its neighboring edge. Let us consider that the surfaces of the image are linear. This is a reasonable assumption if the image is smoothed before edge analysis. The attributes of the step edge are noise, contrast (the cumulative intensity change that occurs across the edge), steepness (the surface slope within the interval, across the profile, in which the bulk of the intensity change occurs), and finally its width (the size of this interval). The relation between these attributes is $steepness \times width = contrast$. The attributes of the double edge are those of each step plus the distance between them (see Fig. 2).

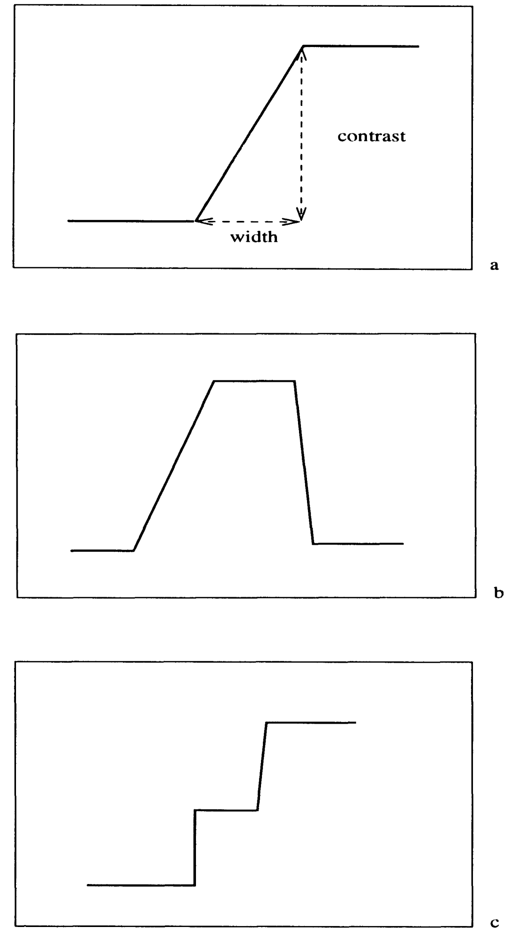

Fig. 2. (a) Step edge profile, (b) pulse edge profile, (c) staircase edge profile.

Edge noise is estimated by using our algorithm proposed in reference (17). The contrast, the steepness, the width, and the distance between the two discontinuities of the double edge are computed by using fitting techniques similar to those in references (22–24). The estimation of all of these attributes takes into account the shape of the edge profile and therefore requires identification of the edge model. For this purpose, we use a set of rules like the following two: (1) A given point is a step edge if its contrast is high and the sum of variances, computed on two elongated bar masks perpendicular to edge orientation at this point (Fig. 3), is smaller than twice the noise energy. (2) A given point belongs to a double edge if the maximum contrast of the two discontinuities is greater than the standard deviation of the noise and the sum of variances is greater than twice the noise energy. This initial classification is improved by taking into account chains (a set of edge points which are filiform and contiguous) and using a vote process.

It should be recalled that the edge is segmented into homogeneous edgels and the attributes of each edgel are computed. An edgel is a set of edge points having homogeneous attributes. The segmentation criteria

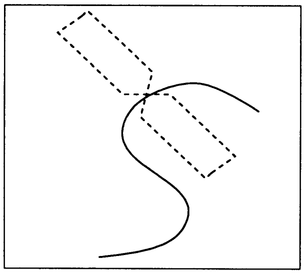

Fig. 3. Elongated bar masks.

are based on geometrical and photometrical attributes. The initial reference edge is segmented by the user into primitives: smoothed curves and straight-line segments. Segmentation of the current reference edge is based on the segmentation of the initial reference edge. After that, the curve-growing procedure[25] is used to segment the primitives according to their photometrical attributes (i.e. edge type, noise, width, contrast, and steepness).

4. DETECTOR CHOICE

Given the set of edgels determined by image structure analysis, the goal of the detector choice is to select an appropriate edge detector and its scale. In general, an edge detector specifies neither the precise context in which it can be successfully used nor the scale computation rule. In practice, it is commonly viewed as a functional $f_s(I(x, y))$ where $I(x, y)$ is the image and s is the scale. We propose to specify an edge detector not only by its algorithm but also by the context in which it is meaningful. The context represents the characteristics of edges on which the detector can be used successfully. Usually, the scale takes its values in an infinite interval. Since close scales produce edges having similar characteristics, we sample the scale and take into account only certain values. The choice of these values depends on the properties of the detector and its implementation method (e.g. convolution masks, IIR filter, FFT). These considerations allow us to specify the knowledge base of the SED system. For this purpose, we have chosen a production system using *if–then* rules as a basic representation framework. The knowledge base is defined as an extensible collection of rules that specify the conditions that must exist for a particular detector to be applied to a particular edge. The production rule is defined as follows:

$$\textit{if}\ (\textit{situation}_1, v_{11}, \ldots, v_{1m})$$
$$\vee \cdots \vee\ (\textit{situation}_n, v_{n1}, \ldots, v_{nm})\ \textit{then}\ (\textit{algorithm, scale}).$$

The condition part of the rule is a representation of the context and is formed by the disjunction of situations. Each *situation* is a conjunction of predicates formed from the edge attributes. v_{ij} is a performance vector for the pair (algorithm, scale) in the ith situation where the edge has an orientation j. The orientation is sampled with period π/m. This performance vector includes the failure probability, delocalization error, and computation time. Failure probability and delocalization error are computed using the quantitative evaluation procedure given in reference (14). The conclusion part of the production rule is formed by the couple (algorithm, scale).

The production rules produce a matching between detector properties and edge characteristics. We described the pertinent characteristics of an edge in the previous section. Some of the properties of a given detector are those of its smoothing filter and its differentiation operator. The smoothing filter properties are the duration of its impulse response, linearity, and invariance to rotation. The differentiation operator properties are linearity, order, and invariance to rotation. There are other properties of an edge detector which cannot be easily deduced from its smoothing filter and differentiation operator, for example, its computational complexity and its goal: the characteristics of the edges which can be localized by the detector (e.g. detector of straight edges, detector of closed edges). It has been pointed out[26] that directional detectors are more suitable in the case of straight-line edges. Since the edge orientation is assumed to be known, we consider two kinds of edge detectors: directional detectors (those which are parameterized by the edge orientation), and general detectors (those which are not destined for a particular edge).

Given the set of edgels, we begin by searching for valid detectors. When we use directional detectors, each edgel is fitted by a set of straight-line segments. For each edgel and for each segment of this edgel the detectors whose condition parts are validated by its attributes are retained. We use a forward chaining algorithm to match rules against edge attributes.[27] Knowledge expressed by these rules describes the mutual influence between edge and detectors. The following examples illustrate the knowledge that we use in SED system:

(1) The Laplacian operator is suitable for the detection of certain junctions, smooth curves, and double edges. This operator is sensitive to scale variation and is suitable for multi-scale edge detection. The gradient operator is suitable for noisy and blurred edges. The directional operator is suitable for straight-line segments.
(2) The performance of the most commonly used detectors is sensitive to edge orientation. The influence of edge orientation on their performance is symmetric at $\pi/4$ and is maximal at this orientation. The performance of these detectors is also affected by subpixel error; i.e., performance decreases as subpixel error increases.
(3) For a step edge, the delocalization and the omission errors of the popular exponential filter

$e^{-\alpha|x|}$ used in the detectors of Shen and Castan[28] are low. The regularizing filter $(a\cos(w|x|) + b\sin(w|x|))e^{-\alpha|x|}$ used in Deriche's detector[29] provides a low delocalization error for a double edge.

(4) The scale of a detector depends both on its properties (in particular, the implementation method used) and on the edge characteristics. The behavior of edges in scale space provides knowledge about the choice of scale. For example, we use smaller scale whenever it yields adequate performance. Noisy and blurred step edges require the use of a large scale. For double edges, the scale must be smaller than the width of the edge to decrease the interaction between the two edges.

After the rules have been fired, we choose the best detector. The selection criteria for the best detector among those being validated are based on the performance and the requirements of the user as to the properties of the final solution. More precisely, we retain the detector that minimizes the linear combination of failure probability $p(d)$, delocalization error $l(d)$, and computation time $t(d)$:

$$c(d) = \alpha p(d) + \beta l(d) + \gamma t(d),$$

where d is a detector. User requirements for the final solution are used to compute the combination scalars α, β and γ. $p(d)$ includes the following three factors: history of the success or failure of d in each situation, its performance as measured by quantitative analysis, and the total length of the edgels which can be detected by the detector d. $l(d)$ includes the delocalization error measured by quantitative analysis. $t(d)$ is the computational complexity of d which is assumed to be independent of the edge. In practice, the number of candidate detectors and the number of edgels are low. Consequently, $c(d)$ is computed for each detector and the one having the lowest value is the best.

5. RESULTS ANALYSIS

In analyzing our results, we consider two aspects. The first concerns measurement of the difference between the initial reference edge and the computed edge to choose the next action to undertake. When the current solution does not meet the requirements of the user, a new reference edge is generated and the feedback loop is run. The new reference edge is made up of those edgels of the initial reference edge which have not yet been found. The second aspect concerns the correction of selection errors resulting from the use of inaccurate *a priori* knowledge. At each loop, the contribution of each detector which was run is updated and falsely selected detectors are discarded.

The results analysis requires a matching process which compares the computed edge and the reference edge to identify similar edgels. To decrease the number of comparisons we propose to combine edge information obtained at the last run of the selected detector with that obtained from previous loops. The combination process consists in identically labeling, with minimum error, an edge resulting from different detectors but originating from the same physical phenomena. The similar edgels are combined to form a single image, to which are added all edgels which are not matched. This image must include all edgels, with a minimum of redundancy and a small delocalization error. These edgels must be filiform and rapidly computable.

We assume that the SED system has run several loops and that the obtained edgels have been combined. Let us consider I_c, the image obtained from the edge combination procedure and I_r the result of the last run of the detector. All edges in I_r which do not correspond to the initial reference edge are suppressed. Identification of similar edgels is done by matching I_r and I_c. The similarity between edges depends on their photometrical and geometrical properties. To take into account the continuation of edgels and to make the process global, we updated this similarity measure using Faugeras's algorithm of stochastic relaxation.[30] The matching process classifies edges into three classes: (1) isolated edges; those belonging only to one image, (2) similar step edges, and (3) similar double edges. We assign to each class an appropriate combination rule. A new image I_n is created and the isolated edges are added. For similar edges, we examine each edgel and we choose the edgel belonging to I_r, or the one belonging to I_c, or produce an edgel that is combination of the two, as shown in Fig. 4. It should be noted that we distinguish between Fig. 4(e) and (f) because with each edge is associated the detector that produces it, and the delocalization error and the computation time of the detectors may be different. It is possible to construct the image I_n by choosing among the four cases without considering neighboring edgels. However, we are interested in the combination procedure which reduces global ambiguity and global delocalization error. Consequently, there are 4^N different solutions to construct the image I_n, where $N = max(p, q)$, p is the number of edgels in I_c and q is the number of edgels in I_r. To avoid a combinatorial solution, we use an A^* algorithm[27] and an heuristic function $c(n)$ which represents the cost of adding the nth edgel. $c(n)$ is recurrent and is defined in terms of three factors: edgel length, delocalization error and the computation time of the detector.

6. EXPERIMENTAL RESULTS

The implemented system contains twenty selection rules related to the two popular step edge detectors derived from Canny's criteria[26] (see appendix). The first of these is Deriche's gradient edge detector (DGD)[29] with a scale α taking its value in the interval $]0, \infty[$. In practice, only the values of the scale α belonging to the interval $[0.75, 1.5]$ are usable, because of the stability and sensitivity of this detector.

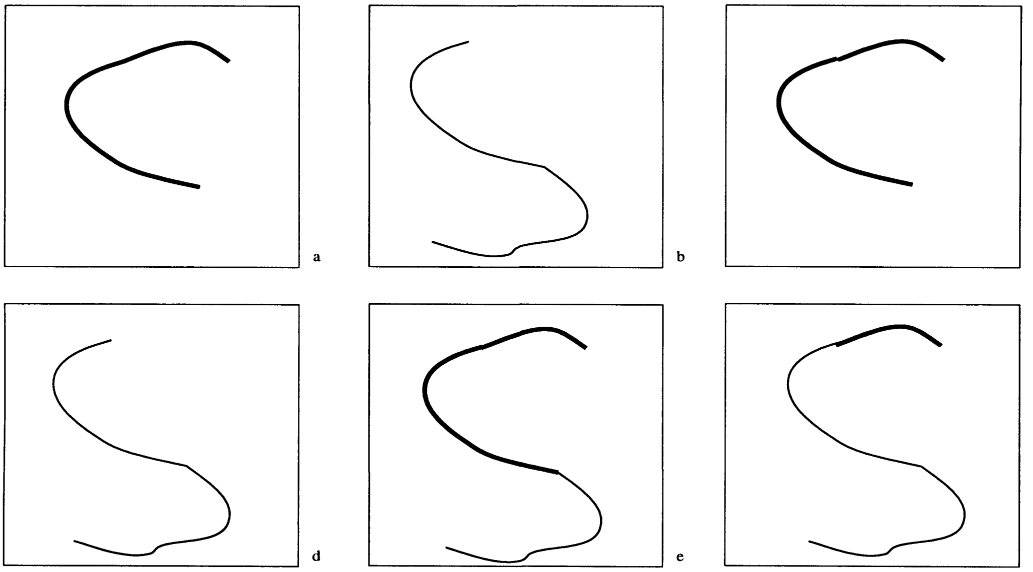

Fig. 4. (a) and (b) Examples of edges to be combined. (c)–(f) are the four possible combinations.

Furthermore, this detector is also used in a particular direction to extract a linear edge, which we call DGDd. Second, we have the Laplacian detector of Shen and Castan (DRF)[28] which has the scale $a \in]0, 1[$. The values of the scale of the DRF detector which were used are in $[0.25, 0.35]$. Note that both scales α and a are inversely proportional to the Gaussian scale (i.e. large scale corresponds to low α and vice versa). To illustrate, we give the following selection rules used in the SED system.

if *the edge is a step* **and** *its signal-to-noise ratio is less than* 2 **then** *use the DGD detector with* $\alpha = 0.75$.

This rule describes a noisy step edge. In this case a first-order differentiation operator is more suitable, because higher-order operators are more sensitive to noise. Strong (higher scale) smoothing is necessary. But to avoid detector stability problems we use $\alpha = 0.75$. Depending on the edge orientation θ, the performance of the $(DGD, \alpha = 0.75)$ given in terms of the delocalization error $l(DGD)$ and the failure probability $p(DGD)$ is as follows: $(l(DGD) = 0.375, \; p(DGD) = 0.251, \; \theta = 0)$, $(l(DGD) = 0.453, \; p(DGD) = 0.252, \; \theta = \pi/8)$, and $(l(DGD) = 0.476, \; p(DGD) = 0.254, \; \theta = \pi/4)$. The performance at a particular orientation which is not given explicitly in the rule can be estimated easily using an interpolation process and the symmetry property. In fact, the performance of this detector is symmetric at $\theta = k\pi/4$, where k is an integer.

if *the edge is a step* **and** *its signal-to-noise ratio is greater than* 6 **and** *its steepness is greater than* 67° **then** *use the DRF detector with* $a = 0.35$ *or the DGD detector with* $\alpha = 1.5$.

This rule describes a well-defined step edge. In this case both detectors can be used at lower scale ($a = 0.35$, or $\alpha = 1.5$). Depending on the edge orientation, the performance of the (DGD, $\alpha = 0.75$) is $(l(DGD) = 0.00, \; p(DGD) = 0.00, \; \theta = 0)$, $(l(DGD) = 0.016, \; p(DGD) = 0.00, \; \theta = \pi/8)$, and $(l(DGD) = 0.142, \; p(DGD) = 0.00, \; \theta = \pi/4)$. The performance of the (DRF, $a = 0.35$) is $(l(DRF) = 0.00, \; p(DRF) = 0.00, \; \theta = 0)$, $(l(DRF) = 0.015, \; p(DRF) = 0.00, \; \theta = \pi/8)$, and $(l(DRF) = 0.075, \; p(DRF) = 0.00, \; \theta = \pi/4)$. Both detectors are implemented using an IIR filter and their computational complexity is scale free. The DRF detector is 3 times faster than the DGD detector.

The integration of other detectors in the SED system requires the definition of their properties using an evaluation process and the formulation of these properties in terms of production rules. The addition of rules to the knowledge base is a simple task. For instance, the gradient of Gaussian,[26] the gradient of Shen and Castan,[28] the Laplacian of Gaussian,[14] and the Laplacian of Deriche[29] have been studied by many researchers and their performance is as well known as the performance of DRF and DGD. Their integration is straightforward.

To validate the proposed approach, let us consider the real image in Fig. 5(a). Figures 5(c) and (b) present the given edge and the window which contains it. To find this edge the SED system run three loops. In the first loop, two edgels are produced by the edge analysis step (Fig. 5(c)). The first edgel is numbered 1; its signal/noise ratio is -2.3 dB and its steepness is 41°. The second is numbered 2 and its attributes are signal/noise ratio 6.5 dB and steepness 76.7°. The validated detectors are: (DRF, $a = 0.25$), (DGD,

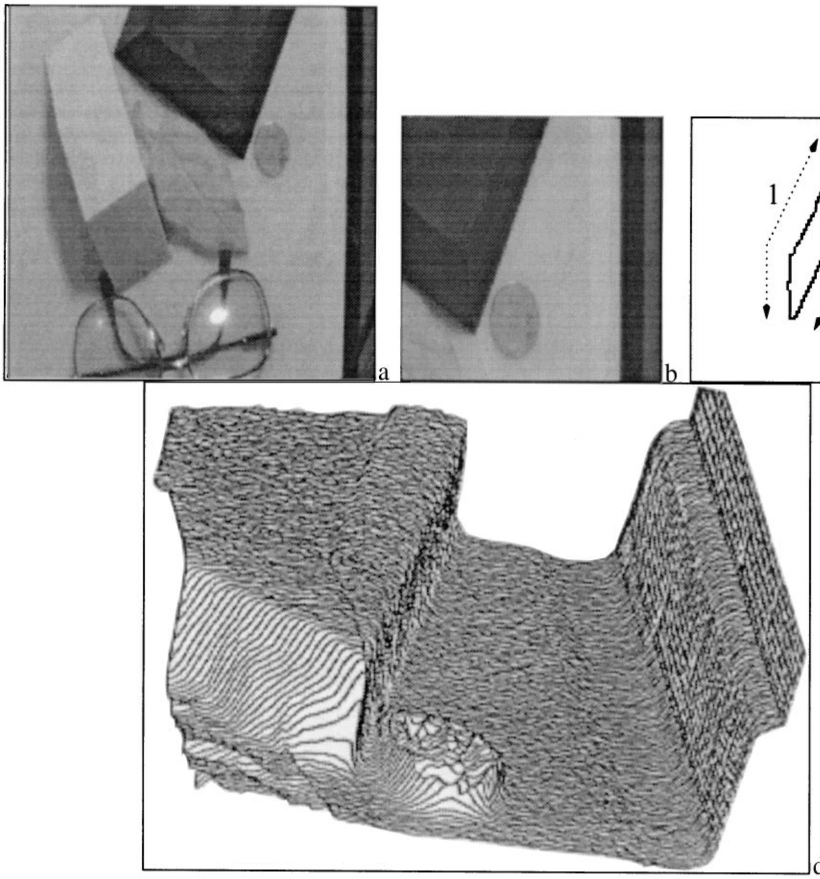

Fig. 5. (a) Original image, (b) a window of the image in (a), (c) the reference edge, (d) image surfaces of (b).

$\alpha = 1$), (DGD, $\alpha = 0.75$), (DRF, $a = 0.3$), (DGD, $\alpha = 1.2$), (DGDd, $\alpha = 1$, $\theta = 6$), and (DGDd, $\alpha = 1.2$, $\theta = 18$). The one retained is (DRF, $a = 0.3$). Figure 6 presents the results of this detector. Note that the given edge is not completely found, so a new reference edge is generated. In the second loop the new reference edge is the one described in Fig. 6. We change the parameter values used in the image structure analysis procedure (e.g. the size of the neighborhood considered when analyzing the edge). The detector retained is (DRF, $a = 0.25$). The edges obtained using this detector are similar to those given by (DRF, $a = 0.3$). Therefore, when combining edge information given by the (DRF, $a = 0.3$) and (DRF, $a = 0.25$) detectors (DRF, $a = 0.25$) is suppressed because it has a greater delocalization error. In the last loop, the Deriche detector with $\alpha = 1$ is selected and its results are given in Fig. 6. The edges given by the DRF and Deriche detectors are combined and the results are given in Fig. 6.

The DRF detector is more efficient than the Deriche detector and its delocalization error is generally lower. Therefore, the obtained solution combines the performance of the DRF detector and Deriche detectors: the computed edge has a small delocalization error, small gaps, and can be computed efficiently. However, study of this example shows that selection error can occur because of imprecise selection rules.

Another result is given in Fig. 7. Figure 7(a) is a SEASAT image of an oil layer on the surface of the sea. In this image, noise is multiplicative and edges are blurred. Laplacian and gradient detectors give edges which are different. Figure 7(b) shows the edges obtained by SED system. This solution combines edges obtained by (DGD, $\alpha = 0.75$) and (DRF, $a = 0.25$).

7. CONCLUSIONS

In this paper we have presented a general approach for selecting edge detectors and automatically computing their scales. By the use of several sources of information (the characteristics of the given edge, the properties of the detectors combined with their results, the influence of edge characteristics and the properties of the detector on its performance), we reduce the search space. Thus, we avoid a combinatorial search, in contrast with the IMPRESS system of Hasegawa *et al.*[2] By using of the selection criteria and the edge combination procedure we combine

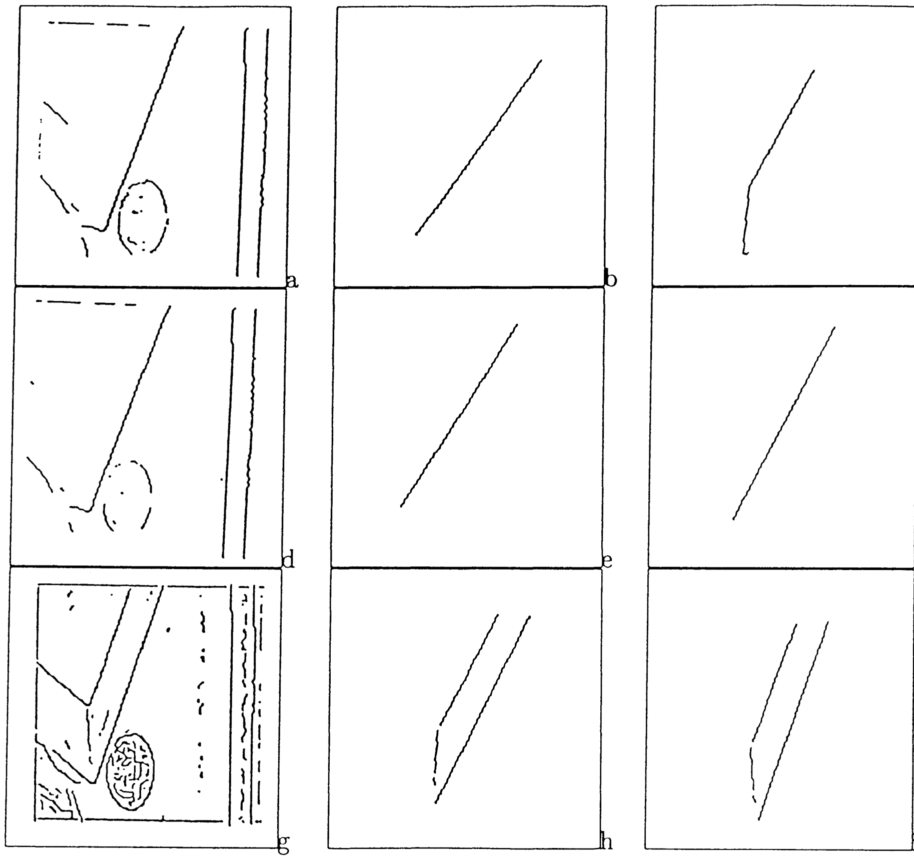

Fig. 6. (a) The edges obtained using (DRF, $a = 0.3$), (b) the results of the elimination process (suppression of noisy edges and those which do not match the given edge), (c) the new reference edge, (d) the edges obtained using (DRF, $a = 0.25$), (e) the results of the elimination process, (f) the edge obtained by combination of the edge information given in (b) and (e), (g) the edges obtained using (DGD, $\alpha = 1$), (h) the results of the elimination process, (i) the edge obtained by combining the edge information given in (f) and (g).

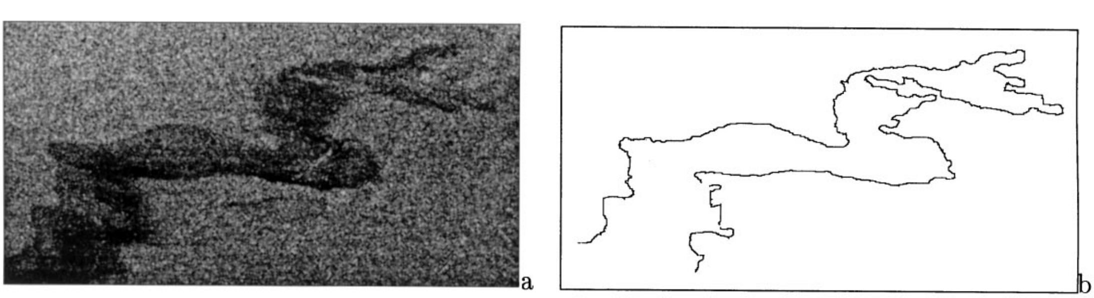

Fig. 7. (a) SAR image of oil spill, (b) the edges obtained using (DGD, $\alpha = 0.75$) and (DRF, $a = 0.25$).

some desirable qualities of edge detectors that are antagonistic: good accuracy, good noise reduction, low computation time. These qualities are essential when performing geometric analysis on the image for the purpose of restoring the three-dimensional structure of the scene.

The main question which can be asked concerns the benefits of this kind of system. There are many; we will give only three examples here. The first is academic: the SED system leads to a better understanding of image structure and edge definition. The second benefit is the possibility of incorporating it into a computer vision system which uses feedback analysis to complete underlying representations.[31] The third benefit is that it can facilitate the use of image processing packages. Although, many catalogs of edge detectors exist, they have merely contained a large, mostly unorganized set of algorithms and provided only very simple tools to assist the user in finding the required one. The nature of the edge detection problem is such that in many cases, an edge detector is almost but not ideally suited for finding all edges in the given image. Whether or not that detector can be used efficiently depends largely on the difficulty of adapting its parameters to the characteristics of the given edge. In the SED system, this is accomplished by introducing the context in which a detector can be used successfully. However, there are limitations in the SED system. In fact, little work has been done on the formulation of the context in which detectors can be used. Existing work on evaluation of detectors has not been done in a way that allows the selection of these detectors and their scales. The knowledge resulting from these detector evaluation schemes is uncertain. To overcome this limitation, we hope to develop a learning procedure for the automatic acquisition of selection rules. Another limitation concerns the use of only two detectors and two types of edges: steps and double steps. The SED system will be enriched by the integration of other detectors and other edge types (i.e., lines, junctions).

Acknowledgements—This work was partially completed while the authors were at Crin-Inria Lorraine (France). We thank G. Giraudon (Inria Sophia Antipolis), G. Masini (Crin-Inria Lorraine) and R. Mohr (Lifia-Imag) for their help.

APPENDIX

This appendix shows some rules related to step edges. Each rule is divided into two parts: a precondition and a postcondition. The precondition is a Boolean expression that must be true before a choice of detectors can be made. The postcondition is an assertion that becomes true when its precondition is satisfied. To shorten the presentation of rules, we have added a disjunction operator to the postcondition. Both preconditions and postconditions are written as well-formed formulae in first-order predicate calculus. We introduce the following predicates:

profile(x, y): edge **x** has as a profile y
value(x, y): the value of x is y
form(x, y): the form of x is y
snr(x, y): the signal/noise ratio of x is y (i.e. very low, low, high, or very high)
steepness(x, y): the steepness of x is y (i.e. very low, low, high, or very high).

Rule 1: if *profile*(edge, step) and (*snr*(edge, low) or *steepness*(edge, low)) then
(*value*(detector, DGD) and *value*(α, 1)) or
(*value*(detector, DRF) and *value*(a, 0.25))

Once a detector and its parameter are chosen, the performance is then estimated according to the edge orientation θ. As an example, in Table A1 we give the performance tables associated to detectors DGD and DRF specified by rule $R1$. As explained in Section 6, the performance is given in terms of the delocalization error l and the failure probability p.

Table A1. Example of performance of DGD and DRF detectors

	DGD performance		DRF performance	
θ	l	p	l	p
0	0.106	0.006	0.075	0.001
$\pi/8$	0.315	0.009	0.123	0.005
$\pi/4$	0.404	0.022	0.139	0.008

Rule 2: if *profile*(edge, step) and (*snr*(edge, very_low) or *steepness*(edge, very_low)) then
(*value*(detector, DGD) and *value*(α, 0.75)).

Rule 3: if *profile*(edge, step) and (*snr*(edge, high) and *steepness*(edge, high) or *snr*(edge, high) and *steepness*(edge, (very_high) or *snr*(edge,very_high) and *steepness*(edge, high)) then
(*value*(detector, DGD) and *value*(α, 1.2)) or
(*value*(detector, DRF) and *value*(a, 0.3)).

Rule 4: if *profile*(edge, step) and *snr*(edge, very_high) and *steepness*(edge, very_high) then
(*value*(detector, DGD) and *value*(α, 1.5)) or
(*value*(detector, DRF) and *value*(a, 0.35)).

Rule 5: if *profile*(edge, step) and *form*(edge, rectilinear) and (*snr*(edge, low) or *steepness*(edge, low)) then
(*value*(detector, DGDd) and *value*(α, 1)).

The performance of the DGDd is orientation free and therefore it is equal to the performance of DGD when the orientation θ is zero.

Rule 6: if *profile*(edge, step) and *form*(edge, rectilinear) and (*snr*(edge, very_low) or *steepness*(edge, very_low)) then
(*value*(detector, DGDd) and *value*(α, 0.75)).

Rule 7: if *profile*(edge, step) and *form*(edge, rectilinear) and (*snr*(edge, high) and *steepness*(edge, high) or *snr*(edge, high) and *steepness*(edge, very_high) or *snr*(edge, very_high) and *steepness*(edge, high)) then
(*value*(detector, DGDd) and *value*(α, 1.2)).

Rule 8: if *profile*(edge, step) and *form*(edge, rectilinear) and *snr*(edge, very_high) and *steepness* (edge, very_high) then
(*value*(detector, DGDd) and *value*(α, 1.5)).

REFERENCES

1. T. Matsuyama, Expert system for image processing: knowledge-based composition of image analysis processes, *Comput. Vision, Graphics Image Process.* **48,** 22–49 (1989).

2. J. Hasegawa, H. Kubota and J. Toriwaki, Automated construction of image processing procedures by sample–figure representation, in *Proc. 8th Int. Conf. on Pattern Recognition*, pp. 586–588, Paris (1986).
3. I. E. Abdou, Quantitative method of edge detection, Technical Report No. 830, Image Processing Institute, University of Southern California (1978).
4. D. Marr and E. C. Hildreth, Theory of edge detection, *Proc. Roy. Soc. London* **B207,** 187–217 (1980).
5. A. Rosenfeld, Image analysis: problems, progress and prospects, *Pattern Recognition* **17**(1), 3–12 (1984).
6. A. M. Nazif and M. D. Levine, Low level image segmentation: an expert system, *IEEE Trans. on Pattern Anal. and Mach. Intell.* **6**(5), 555–576 (1984).
7. V. Torre and T. A. Poggio, On edge detection, *IEEE Trans. on Pattern Anal. Mach. Intell*, **8**(2), 147–163 (1986).
8. F. Bergholm, Edge Focusing, *IEEE Trans. Pattern Anal. Mach. Intell.* **9**(6), 726–741 (1987).
9. E. P. Lyvers and O. R. Mitchell, Precision edge contrast and orientation estimation, *IEEE Trans. Pattern Anal. Mach. Intell.* **10**(6), 927–937 (1988).
10. Y. Lu and R. V. Jain, Behavior of edges in scale space, *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(4), 337–356 (1989).
11. J. S. Chen and G. Medioni, Detection, localisation, and estimation of edges, *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(2), 191–198 (1989).
12. D. J. Williams and M. Shah, Edge contours using multiple scales, *Comput. Vision, Graph. Image Process.* **51,** 256–274 (1990).
13. P. E. Danielsson and O. Seger, Rotation invariance in gradient and higher order derivative detectors, *Comput. Vision, Graph. Image Process.* **49,** 198–221 (1990).
14. S. Venkatesh and L. J. Kitchen, Edge evaluation using necessary components, *CVGIP: Graph. Models Image Process.* **54**(1), 23–30 (1992).
15. V. Lacroix, Edge detection: what about rotation invariance? *Pattern Recognition Lett.* **11,** 797–802 (Dec 1990).
16. D. Ziou and R. Mohr, An experience on automatic selection of edge detectors, in *Proc. 11th Int. Conf. on Pattern Recognition*, pp. 586–589, Netherlands (1992).
17. D. Ziou, Noise estimation from step edge operator responses, in *Proc. 8th Scandinavian Conf. on Image Analysis*, pp. 1397–1402 (1993).
18. D. Ziou and S. Tabbone, A multi-scale edge detector, *Pattern Recognition* **26**(9), 1305–1314 (1993).
19. D. Ziou and J. P. Fabre, Effects of edge orientation on the performances of first order operators, *Pattern Recognition Lett.* **15,** 1053–1063 (1994).
20. S. Tabbone and D. Ziou, On the behavior of the Laplacian of Gaussian for junction models, in *2nd Ann. Joint Conf. on Information Sciences*, pp. 304–307, NC, USA (1995).
21. D. Ziou and S. Wang, Isotropic processing for gradient estimation, In *Proc. IEEE, Int. Conf. on Computer Vision and Pattern Recognition*, pp. 660–665, San Francisco (1996).
22. R. M. Haralick, Second directional derivative zero-crossing detector using the cubic facet model, in *Proc. 4th Scandinavian Conf. on Image Analysis*, pp. 17–30, Trondheim (1985).
23. V. S. Nalwa and T. O. Binford, On detecting edges, *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(6), 699–713 (1986).
24. Y. G. Leclerc and S. W. Zucker, The local structure of image discontinuities in one dimension, *IEEE Trans. Pattern Anal. and Mach. Intell.* **9,** 341–355, (1987).
25. S. W. Zucker, 'Survey': Region growing: childhood and adolescence, *Comput. Vision an Image Process.* **5,** 382–399 (1976).
26. J. F. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(6), 679–698 (1986).
27. N. J. Nilsson, *Problem-Solving Methods in Artificial Intelligence*. McGraw-Hill, New York (1971).
28. J. Shen and S. Castan, An optimal linear operator for step edge detection, *CVGIP: Graph. Models Image Process.* **54**(2), 122–133 (1992).
29. R. Deriche, Fast algorithm for low-level vision, *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(1), 78–87 (1990).
30. O. D. Faugeras and M. Berthod, Improving consistency and reducing ambiguity in stochastic labeling: an optimization approach, *IEEE Trans. Pattern Anal. Mach. Intell.* **4,** 412–423 (1981).
31. M. Nagao, Control strategies in pattern analysis, *Pattern Recognition* **17,** 45–56 (1984).

About the Author—D. ZIOU was born in Algeria. He received an Engineering Degree in Computer Science from the University of Annaba (Algeria) in 1984, and the Ph.D. degree in Computer Science from the Institut National Polytechnique de Lorraine (INPL), France in 1991 respectively. From 1987 to 1991 he served as a Teaching Assistant at the Universities of Metz and Nancy, France. During the same period, he was a researcher in the Centre de Recherche en Informatique de Nancy (CRIN) and the Institut National de Recherche en Informatique et Automatique (INRIA) in France. From 1992 to 1993, he served as lecturer in a school of computer engineering (EERIE), Nimes in France. Presently, he is Associate Professor at the university of Sherbrooke in Canada. His research interests include image processing, computer vision and pattern recognition.

About the Author—A. KOUKAM received the Ph.D. degree in computer science from the Unversity of Nancy I, France in 1990, where he served as a Teaching Assistant (1986–1989), and a researcher in the Centre de Recherche en Informatique de Nancy (CRIN, 85-90). Presently, he is Associate Professor of computer science at Institut Polytechnique de Sevenans, Belfort, France. He heads a research project on modeling and analysis of discrete event systems, including software engineering and multi-agent systems.