Merging system for multiscale edge detection

Olivier Laligant
Frédéric Truchetet, MEMBER SPIE
Johel Miteran
Patrick Gorria
IUT
12 rue de la fonderie
71200 Le Creusot, France
E-mail: o.laligant@iutlecreusot.u-bourgogne.fr

Abstract. We present a multiscale edge detection algorithm whose aim is to detect edges whatever their slope. Our work is based on a generalization of the Canny-Deriche filter, characterized by a more realistic edge than the traditional step shape edge. The filter impulse response is used to generate a multiscale edge detection scheme. For the merging of the edge information, we use a geometrical classifier developed in our laboratory. The segmentation system thus set up, after the training phase, does not require any adjustment or depend on any parameter. The main original property of this algorithm is that it leads to a binary edge image without any threshold setting. The quality of the results is inferior to that for classical multiscale merging approaches; nevertheless, this system, studied for real-time functioning, presents satisfactory performance for well-contrasted images and excellent performance for noisy images. © *2005 Society of Photo-Optical Instrumentation Engineers.*
[DOI: 10.1117/1.1869998]

Subject terms: blurred and noisy images; edge detection; multiscale analysis; multiscale merging; geometrical classification.

Paper 030586 received Nov. 21, 2003; revised manuscript received Aug. 30, 2004; accepted for publication Oct. 18, 2004; published online Mar. 10, 2005.

1 Introduction

Edge detection is one of the basic operations in all artificial vision systems (shape recognition, character recognition, defect detection on industrial parts, etc.). The edges, or *contours*, are characterized by sharp or gradual variations of luminance in the image. The detection and localization of these variations in luminance have been the subject of numerous publications. Gradient estimations (Roberts's gradient, for instance) are the classical methods, which give satisfactory results in the case of slightly noisy images. In the case of noisier images, researchers have developed different optimal operators associating low-pass filtering with the gradient detector.[1–3] It seems that the path initiated by Canny[4–6] leads to the best results in the space of linear filtering methods. In this approach a step edge model is considered, and the impulse response of the filter is found by the maximization of three criteria: signal-to-noise ratio, localization, and multiple-response suppression in the filtered image. However, a difficulty remains: the impulse response function obtained depends on a scale parameter, and the performance is very sensitive to the choice of this parameter. If the edges to be detected are sharp, a narrow filter can be chosen, but if the edges are diffuse, a larger one will give more satisfactory results. In the real-world 3-D images that we have to deal with in our current research work, the edge blur due to lack of focusing is not homogeneous, and we have to find a trade-off for the scale factor. In order to solve this problem we have tried to merge the elements of a family of multiscale edge images. Such an approach has been previously tried by some authors.[7–11]

The method proposed here is based on a statistical approach. We should note that Forthoffer et al.[12] used this approach with Haar decomposition (which is not the best for edge detection) and a neural network classifier.

We present in the second section a generalization of the Canny-Deriche filter for ramp edge detection.[13] The operator involved in the segmentation step of the algorithm is detailed in the third section with special emphasis on the classifier. In section 4 we present the global system and its properties. Some results on real images show the limitations and advantages of the system. In the concluding section we summarize the main results and introduce some of our current work in the field.

2 Multiscale Edge Detection

2.1 *Edge Model and Edge Detector*

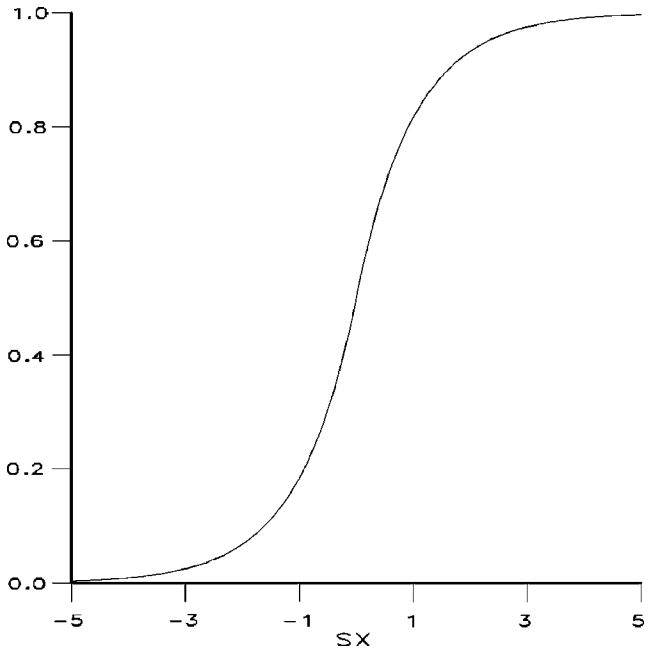
The acquisition of a real-world image by a camera through an optical device entails the presence of a certain blur, which is generally well modeled by the effect of Gaussian filtering on the image. The result for the edge is an erf-type law, and explicit calculations for optimization criteria appear to be impossible. Furthermore, the blur of real edges can have other causes than optical ones (the movement of an object in the scene, weak resolution of the image, diffuse illumination, etc.). Thus, this model is unable to represent every aspect without approximation. For these reasons we choose the edge model (see Fig. 1) introduced by Petrou and Kittler.[14]

This function is a first-order approximation of the majority of edges encountered in real image processing and corresponds to the filtering of an ideal edge by a low-pass filter defined by its impulse response:

$$t(x)=\frac{s}{2}\exp(-s|x|). \tag{1}$$

The construction of the edge detection filter has been achieved following Canny's approach and taking into ac-

0091-3286/2005/$22.00 © 2005 SPIE



$$C(x)=\begin{cases}1-\frac{e^{-sx}}{2} & x\geq 0\\ \frac{e^{sx}}{2} & x<0\end{cases} \quad (23)$$

Fig. 1 Edge model defined as an exponential function with $s>0$.

count a smooth profile of the edge to be detected.[13] The optimization criteria are those used by Canny[4,15] (signal-to-noise ratio, localization, and multiple response).

The impulse response defined by Eq. (1) is used to construct a normalized filter [Eq. (2)]:

$$f(x)=\begin{cases}N\sqrt{s}(ksxe^{msx}-e^{msx}+e^{sx}), & x<0,\\ N\sqrt{s}(ksxe^{-msx}+e^{-msx}-e^{-sx}), & x\geqslant 0.\end{cases} \quad (2)$$

Here N is the normalization factor to make $|\int_{-\infty}^{0}f(x)\mathrm{d}x|=1$.

The recursive implementation of the filter gives a stable third-order filter, and its performance shows an increase of the signal-to-noise ratio of about 20% in the case of blurred and noisy images in comparison with the Deriche filter.[16]

The scale parameter s must thus be chosen with respect to the content of the image and more particularly with respect to the finest edge. Table 1 shows the width of the edge with respect to scale s.

An initial scale $s_0=6$ is chosen for the rest of this work. This choice can obviously be modified for less abrupt edge profiles. The coarsest scales will be fixed by the multiscale decomposition. The factor m sets the width of the filter for a given scale s. The parameter k is a shape factor; for a given pair m, s it acts as a filter decrement (localization) in Fourier space. In this work $m=0.2$ and $k=0.3$. Discussions on the choice of these parameters can be found in Ref. 17.

Table 1 Width ℓ of the edge model for scale s. Here ℓ is defined by $|f(\ell/2)-f(-\ell/2)|\approx 0.95$.

s	ℓ
1	6
2	3
3	2
6	1

2.2 *Multiscale Detection*

The length of the filter being infinite, we have implemented its calculation in a recursive way. This leads to third-order recursive infinite impulse response (IIR) filters. The generalization to two dimensions is based on separable filtering with an integral filter as the regularization filter (more details can be found in the work of Bourennane et al.[13]). The algorithm is given in Sec. 6. The monoscale algorithm has been implemented in an application-specific integrated circuit (ASIC). The multiscale one has been programmed in field-programmable gate array (FPGA) circuits.[18] Both the circuits work at our current standard video rate (25 images/s).

The edge signal modulus is given by the following expression with the integral function h of the filter f for a scale p:

$$G_p(x,y)=[Gx_p(x,y)^2+Gy_p(x,y)^2]^{1/2} \quad (3)$$

with

$$Gx_p(x,y)=f_p(x)h_p(y)*I(x,y), \quad (4)$$

$$Gy_p(x,y)=f_p(y)h_p(x)*I(x,y), \quad (5)$$

$$s=\frac{s_0}{a^p} \quad (a>0). \quad (6)$$

In this work the classical value $a=2$ is chosen. In actual fact, for the merging system the Manhattan modulus is used and we thus exploit the following luminance information:

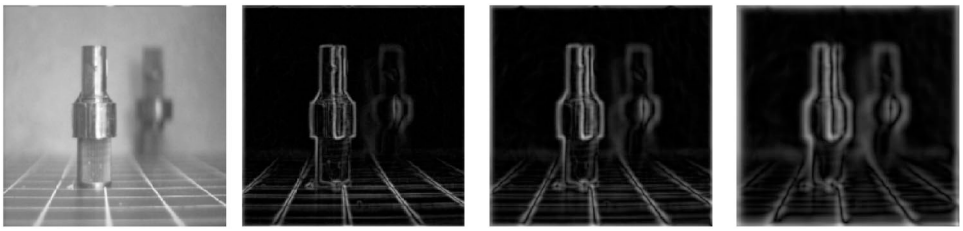


Fig. 2 Original image and modulus of wavelet coefficients for $s=6$ with $p=0$, $p=1$, and $p=2$ and x, $y \in [0,255]$.

$$l^p_{x,y} = l^p(x,y) = |Gx_p(x,y)| + |Gy_p(x,y)|. \quad (7)$$

We present in Fig. 2 an example of results where the complementarity between the different scales appears clearly. In the upper scale image $(p=0)$, the sharp edges are well detected and localized, but the blurred edges are poorly detected, and to find them we have to search in the lower scale image $(p=2)$. Therefore it seems interesting to merge the different scale images. For the merging operator a classifier operating at video rate after supervised training has been chosen.

3 Merging Operator

3.1 *Introduction*

The different scales of the decomposition data have to be merged. The fine scales contribute to the detection of sharp edges, but are too sensitive to noise and insensitive to blur. On the contrary, the coarse scales eliminate noise and allow blurred-edge detection. We tackle the multiscale segmentation problem by a statistical approach. A classifier is fed with the multiscale edge detection, and the result is a unique binary image.

In order to merge the data produced on different levels, we use a geometric classifier that was developed at our laboratory[19] and that has given rise to a real-time application development in the form of a specific circuit. During the training phase this classifier divides the parameter space into orthogonal hyperplanes. The performance thus deviates from the Bayesian optimum, but the decision time is greatly reduced. This type of classifier is indeed well adapted to quality control applications because it is fed with *a priori* data. These data confer high robustness on the classifier. The difficulty is to label the data properly and to choose their representation space correctly. In the next subsection we present briefly the classification algorithm used.

3.2 *Merging Operator or Classifier*

We are only interested here in classification of pixels (of images or any objects) with respect to their characteristic parameters and not their origins, nor even the validity of the parameters. The more pertinent the attribute vectors are, the more reliable and the faster the decision will be, whatever the correlations among the parameters. We assume that a large number of samples of each class are available during the training phase; in our application that is clearly the case. The method presented here aims at dividing the attribute space into a small set of hyperrectangles called stress polytopes, where each hyperrectangle contains samples of a single class and where the stress is produced by the other class(es). Thus, the classification of a new point may be easily obtained by simple comparators. The implementation of classification by training requires two distinct steps: training and decision.

3.2.1 *Training phase*

The training phase consists in collecting a set $Eq = (e_1, e_2, \ldots, e_q)$ of the most representative samples from the various classes and associating with each sample a local stress. Each sample e_i is defined by an attribute vector $(x_1, x_2, \ldots, x_N)$ in an N-dimensional space and by its corresponding class $C(e_i)$. In practice, in the case of image segmentation, the number of available samples can be equal to the number of pixels of the reference images (several thousands), and the dimension of the attribute space may be very high (5 to 20 parameters).

A region is associated with each sample, such that any point within this region is considered as pertaining to the class of the sample. This local stress polytope guarantees the correct reclassification of the sample as well as of its neighboring points. We determine for each point e_i a hyperrectangle $H(e_i)$, which constitutes a proximity stress to every element from E_2 not belonging to the class $C(e_i)$. The frontiers of this local stress polytope are obtained by seeking, in all directions from $C(e_i)$, the nearest neighbors from different classes. In order to obtain a stress polytope defined by a hyperrectangle, the distance used for the search for the nearest neighbors[20] is $d_\infty(x,y) = \max_{k=1,\ldots,N}|x_k - y_k|$. Let d_p be the distance from the sample to the nearest neighbor so defined; the frontier is then located at a distance $d_f = d_p R$ from the sample. Figure 3 shows a sample e_i and the associated hyperrectangle $H(e_i)$ for a ratio $R=0.4$.

This ratio R characterizes the degree of resemblance a point must meet to be classified in the point class generating this stress. If R tends to zero, then the local stress is

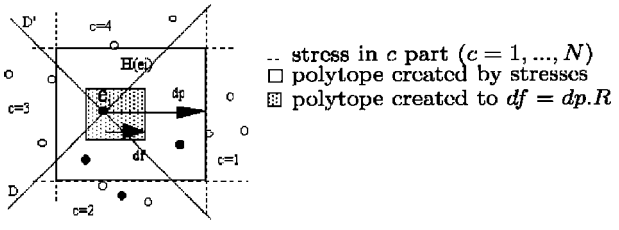


Fig. 3 Hyperrectangle $H(e_i)$ attached to sample e_i.

reduced to the only point generating it; if R tends to 1, the fluctuation is maximum, except for the points belonging to different classes. As an object should not be classified in any case in two distinct classes at a time, the ratio R should remain less than or equal to 0.5 for a multiclass problem. For problems involving only two classes, generally one class is represented by polytopes and there is no condition on R.

Once the set of local stress polytopes for each attribute vector is obtained, the polytopes are merged into larger hyperrectangles while preserving the disjointness of these new polytopes with those from different classes.[21]

The merging ratio (the number of polytopes before merging divided by the number after merging) must increase with decreasing ratio R. If because of integration difficulties it is required to reduce the number of hyperrectangles, it is advisable to choose a small ratio, even to the detriment of the tolerance on the attributes.

Only this new set of polytopes is used during the decision phase. It consists of a number of hyperrectangles h, also called stress polytopes.

3.2.2 *Decision phase*

The decision phase consists in allocating a class to a new attribute vector. The cutting of the attribute space was only done with hyperrectangles defining the characteristic space of a class. The orthogonal boundaries allow a simultaneous processing of images. Let P be a new sample to be classified. This new object will be classified in the class C_p if and only if it belongs to a hyperrectangle H_i of class at least C_p. The verification of the appurtenance of a point M to a class C_l results in a set of comparisons to be done simultaneously, on each parameter, for every hyperrectangle of class C_l. If these operations are done in parallel, the processing time of the decision phase is equal to the time required to do a simple comparison.

In order to obtain a classification system operating at the pixel rate (that is, below 125 ns per decision), we have integrated this operator in the form of an integrated circuit.[22]

3.2.3 *Classifier performance*

We chose two Gaussian distributions, following a procedure similar to that used by Mattews and Hearne,[23] to characterize the performance of the Riffle classifier. We varied the different characteristic parameters (standard deviation σ, average m) so as to compare the performance of this method with the classification optimum. For a given configuration the optimum classification error cannot be less than the intersection area between the two Gaussians. By separating the Gaussian means m_0 and m_1 by amounts $n\sigma\,(n=0,...,4)$, we obtain the curve giving this optimum. In the same way, we moved the Gaussian m_0, calculating at each step the theoretical error probability. Both curves are shown in Fig. 4. One can see that when the parameters are sufficiently discriminant, the error probability tends to the optimum.

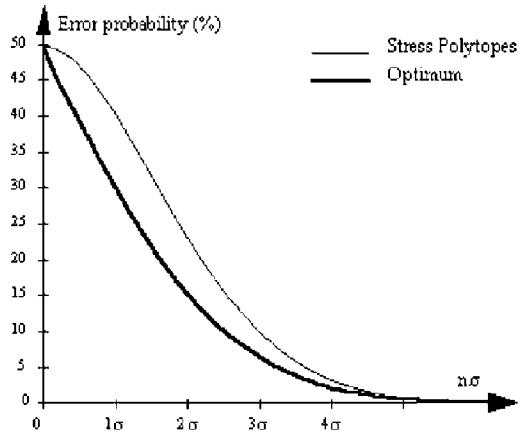


Fig. 4 Classifier performance.

4 Merging System

4.1 *Introduction*

The merging system is composed of a multiscale edge detector, an attribute vector extractor, and a stress-polytope classifier. Both the edge detector and the classifier have been implemented in hardware using ASIC technology, and the data flowing between the various parts are 8-bit coded integers.

The classifier needs an off-line training stage before any on-line operation. This training has to be done once for all, assuming that we process a family of images with common edge features. Figure 5 illustrates this training stage. A training image with well-defined edges is segmented using the edge detector (on the scale corresponding to the mean edge slope), and the result is used as the reference edge image. In parallel this image is blurred and noised before being processed by the multiscale edge detector and sent to the classifier set in training mode. The reference image gives to the classifier for each pixel (represented by an attribute vector) the corresponding class: edge point or not. The attribute vector components are luminance value and combinations of luminance values of neighboring (in space and/or in scale) pixels. The classifier is therefore able to construct the partitioning of the attribute space into hyperrectangles. The decision stage, which is an on-line operation, is illustrated in Fig. 6. The attribute vector is extracted for each pixel from the multiscale edge detector in the same way as in the training stage.

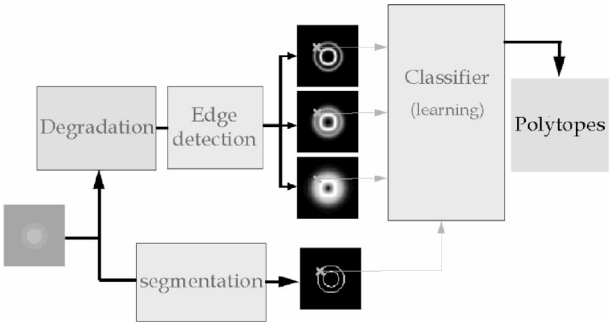


Fig. 5 Training system.

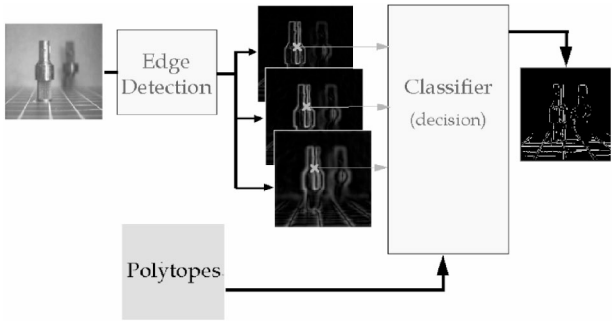


Fig. 6 Merging system.

We present in the next subsections the extraction of the attribute vector and the methods used to evaluate the system performance. Some results are then described to show monoscale and multiscale edge detection on blurred and noised images and on micrographic images, which are hard to process in a classical way without ad hoc parameter setting.

4.2 *Attribute Vector and Performance Evaluation*

4.2.1 *Attribute vector*

Assuming that edge pixels are spatially connected, we have chosen to extract the attribute vector components from linear combinations of points close to the pixel under consideration, these points being taken at the output of the multiscale edge detector.

Let us adopt the following notation: $l^p_{x,y}$ is the luminance of the pixel of coordinates (x,y) relative to the point under consideration at scale p; χ^p is subset of components at scale p. Some examples of these subsets for horizontal and vertical edges are presented in Table 2.

We give in Table 3 examples of attribute vectors. One of them has been used for the images of Fig. 8.

4.2.2 *System parameters*

The system parameters, which are used only during the training stage, are as follows:

- training image
- detector parameter: s
- classifier parameter: ratio R
- training parameter: ratio T, given by

$$T=\frac{\text{number of points belonging to the background class}}{\text{number of points belonging to the edge class}}.$$

Table 2 Example of basic attributes.

$d^p_{\vert}$	$\{127+l^p_{0,0}-l^p_{0,-1}\,;127+l^p_{0,0}-l^p_{0,1}\}$
d^p_{-}	$\{127+l^p_{0,0}-l^p_{-1,0}\,;127+l^p_{0,0}-l^p_{1,0}\}$
$h^p_{\vert}$	$\{l^p_{0,-1}\,;l^p_{0,0}\,;l^p_{0,1}\}$
h^p_{-}	$\{l^p_{-1,0}\,;l^p_{0,0}\,;l^p_{1,0}\}$
$d^p_{/}$	$\{127+l^p_{0,0}-l^p_{1,-1}\,;127+l^p_{0,0}-l^p_{1,-1}\}$
$d^p_{\backslash}$	$\{127+l^p_{0,0}-l^p_{1,1}\,;127+l^p_{0,0}-l^p_{-1,-1}\}$

Table 3 Parameters and performances for edge luminance information $l^p_{x,y}$ with $p=0$. Here P is the percentage of false detection [$P = 100\times$(number of false detections)/(number of true edge points)], and ND the percentage of nondetected points.

Vector	ND (%)	P (%)
$\{d^p_{\vert}\,;d^p_{-}\,;l^p_{0,0}\}$	22.1	23.1
$\{d^p_{\vert}\,;d^p_{-}\,;l^p_{0,0}\,;d^p_{\backslash}\,;d^p_{/}\}$	26.8	14.3
$\{d^p_{\backslash}\,;d^p_{/}\,;l^p_{0,0}\}$	21.0	18.6
$\{d^p_{\backslash}\,;d^p_{/}\}$	18.6	21.1

Once the training has been done, the system can work autonomously. Therefore, in the case of production survey of manufactured parts, the inspection can be performed online as the part passes by. This inspection can be used in order to detect misalignment, aspect defects, faults in size, etc. The system should be very robust and should be capable of dealing with unforeseen situations, such as illumination variations, which can occur in real situations.

4.2.3 *Performance evaluation*

In order to evaluate the system performance we make use of two criteria. The first one is simply the figure of merit proposed by Pratt, which measures the performance of an edge detector on a synthetic image of a sharp straight edge. The second one allows performance estimation for real-world images. It is based on two figures (Table 3): the percentage ND of undetected edge points and the percentage P of false detection (parasitic points). These percentages are computed with respect to the number of edge points detected by a classical method (detection, local-maximum extraction, and hysteresis thresholding). These two figures are presented in a 2-D plane curve obtained by varying the system parameters (see examples in Figs. 7 and 13).

Remark. In order to take into account the constraints linked to hardware implementation, we have measured the performance of the system by software simulation with 8-bit coded data for intermodule exchanges. For the reference classical method, simulations have been done with real floating-point data.

4.3 *Monoscale Evaluation*

4.3.1 *Introduction*

The segmentation scheme presented in the previous sections can be used in a monoscale approach. Its main advantage over the classical method is that it leads to easy real-time implementation and it does not need any ad hoc parameter setting (the training elements are fixed once for all). We present in this section some results obtained with this method. This study, showing the influence of the training parameters, has guided us in our choices for the multiscale scheme. We also see the limitations of the monoscale approach and then, naturally, introduce the multiscale one that we present in the next sections.

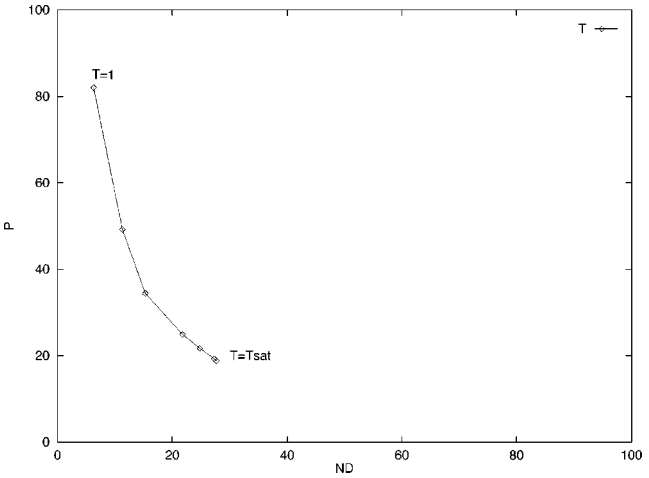


Fig. 7 Graph of P versus ND. Evolution of performance according to the ratio T [$T \in \{1,2,3,4,5,6,7\}$, $s=6$, $R=0.5$; $\{d^p_{\uparrow}, d^p_{-}, I^p_{0,0}\}$ as in Eqs. (10) and (11)].

4.3.2 *Training elements*

In this stage we have three points to address:

- definition of the training image(s)
- setting of the ratio T
- choice of the attribute vector components.

The chosen scale must be approximately the finest scale of the edges to be detected.

For the training image two solutions have been considered. The first one makes use of a real-world image chosen in accordance with the family of images on which the system will be used; in the second one a ''universal'' training image with simple geometrical features has been synthesized. In both cases we have set the ratio T by choosing the best trade-off between P and ND (see Fig. 7). We observed that there exists a saturation point for T beyond which the performances decrease, and

$$T_{\text{optimal}} \in \left[\frac{T_{\text{saturation}}}{2}, T_{\text{saturation}}\right] \approx [3,6],$$

where $T_{\text{saturation}} \approx 6$. Also, it is to be noted that, whatever the training images, the classifier parameter R can be set to $R = 0.6 \pm 0.1$ with good results, so that its setting does not seem to be critical.

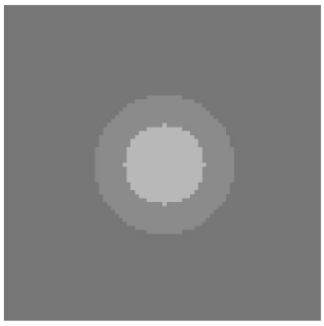


Fig. 9 Training image: two disks of radii 20 and 35 and of amplitudes (relative to background) 50 and 20.

An example of a real-world training image, segmented in a classical way (CW) and then segmented by our classifier (CC), is given in Fig. 8. In this series of trials we have selected on the CW a training zone, which is used as reference for the polytope construction. CC presented the results of the segmentation obtained for the whole image after such a training. The result is not so bad, but the training obtained from the real-world image is generally too special, and if we try to improve its generality by increasing the training zone, we are reduced to training ''by heart'' (more than 300 polytopes!), which gives very poor results when the classifier is then used on another image.

We have solved this problem with the use of a synthetic universal training image. After a lot of trials and in accordance with the spatial invariance requirement, we propose an image constituted of two concentric disks of different radii and different luminances, the edges being assumed to be on the circular boundaries of the disks. An example is given in Fig. 9. We have been led to this choice by the fact that the training with one disk was too poor for the highest-contrast zone of the image, and a good solution has been to add a *generalization disk* with higher luminance.

For the attribute vector components we have been looking for the most concise and robust set of attributes. Too small a number of components does not allow good enough discrimination between the distribution modes; too large a number leads to poor generality of the training, the extreme being learning by heart. The best results have been obtained with the following set, in which the d components are sensitive to the transitions and the Σ ones give a local threshold:

Fig. 8 From left to right: original image; image segmented in the classical way (CW); image obtained by our system (CC) with parameters $\{d^p_{\uparrow}, d^p_{-}, I^p_{0,0}\}$, $p=0$, modulus approximation of the gradient, and real-world training image.

Fig. 10 From left to right: original image; image segmented by extraction of local maxima and hysteresis thresholding (upper threshold=20); image segmented by the classifier. The training has produced 86 polytopes.

$$\{(d_{|}^{p}, d_{-}^{p}, \Sigma_{5}^{p})_{p\in[0,1,\ldots]}\} \tag{8}$$

with

$$\Sigma_{5}^{p} = (l_{0,0}^{p} + l_{-2^p,0}^{p} + l_{2^p,0}^{p} + l_{0,-2^p}^{p} + l_{0,2^p}^{p})/5, \tag{9}$$

$$d_{|}^{p} = \{127 + l_{0,0}^{p} - l_{0,-2^p}^{p}; 127 + l_{0,0}^{p} - l_{0,2^p}^{p}\}, \tag{10}$$

$$d_{-}^{p} = \{127 + l_{0,0}^{p} - l_{-2^p,0}^{p}; 127 + l_{0,0}^{p} - l_{2^p,0}^{p}\}. \tag{11}$$

Some other choices for Σ^p (summation over the four diagonal neighbors, average value of the eight-neighbors) have been found to be nearly equivalent to the mentioned choice.

The results obtained with such an attribute vector and with the so-called synthetic universal training image are presented through two examples given in Figs. 10 and 11. The results are rather more satisfactory than those of the classical method (CW). One has to keep in mind that we do not have any threshold setting to do, so that it is really an unsupervised method.

4.4 *Conclusion on the Monoscale Approach*

Therefore, although this method cannot be compared to a supervised thresholding scheme, if one is interested in a real-time unsupervised edge detection system, the presented monoscale scheme has to be considered. The limitations of this monoscale approach can be seen when it is used on noise-corrupted images or on images with variously blurred edges. For these cases we cannot expect any improvement over the classical way with local maximum detection followed by hysteresis thresholding. That is why we propose now the extension to a multiscale classification.

In this perspective, the study of the monoscale system gives interesting information:

- The parameters of the classifier are fixed as following:

$$R = 0.6 \pm 0.1, \tag{12}$$

$$T \in \{3,4,5,6\}. \tag{13}$$

- The attribute vector (8) leads to good results.
- An efficient training is achieved by the choice of the training image in Fig. 9.

4.5 *Multiscale Classification System*

4.5.1 *Introduction*

As previously said, the use of only one scale does not lead to good detection of the variously contrasted edges that are to be found in a difficult image of the real world. If the chosen scale fits fine edges well, blurred ones will be undetected or ill detected, the quantization on an integer range of the gradient values leading to degradation. If the chosen scale is wide, the gradient estimation will be robust against noise,[24] but the fine contours will be poorly detected, with or without noise. Then multiscale merging seems to give the most complete representation. It must allow a segmentation of the attribute space in good accord with the blurred

Fig. 11 From left to right: original image; image segmented by extraction of local maxima and hysteresis thresholding (upper threshold=20); image segmented by the classifier. The training has produced 86 polytopes.

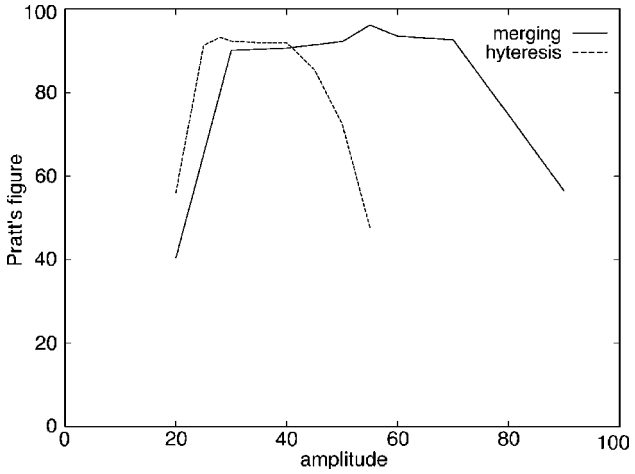


Fig. 12 Variation of Pratt's figure of merit with amplitude of training image and hysteresis threshold (CW). In the particular case of Pratt's image, the best classification is obtained for training with the same standard deviation of the Gaussian noise. The segmentation is indeed optimal for one type of edge only. The merging system is more robust to changes in amplitude.

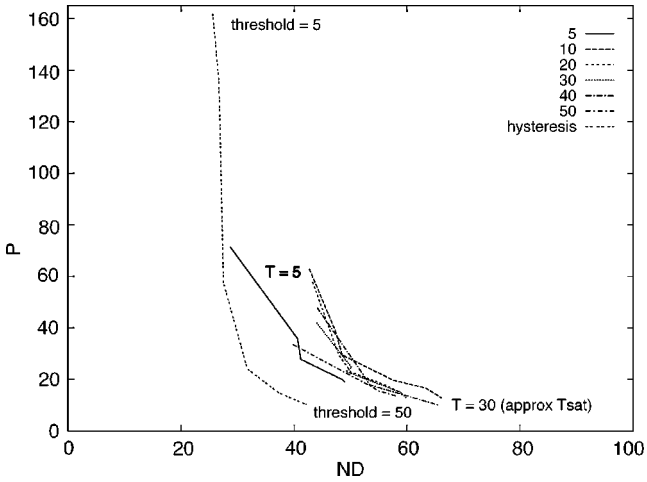


Fig. 13 Performance curve for segmentation by hysteresis on the first scale and merging of the image "Office" embedded in noise of standard deviation 22. The merging curves are obtained for variable T with different amplitudes of the training image, and the hysteresis curve is obtained according to the upper threshold (5,10,...). One can see that in spite of a decrease of performance, the robustness of the merging is better with regard to the tuning. Therefore the choice of amplitude in training is less important than the tuning of hysteresis. The scale used for thresholding is the first one. The curve obtained with second scale spans the same interval as the set of curves obtained for the merging, but it goes much higher.

edges as well as with the fine edges. In order to obtain such a multiresolution scheme, we have to define a multiscale attribute vector.

4.5.2 *Multiscale attribute vector*

After a lot of trials, we have concluded that only three scales are needed to obtain the best results for an image size of 256 by 256 pixels. So the attribute vector should be [see definitions in Eqs. (9), (11), (10)]

$$\{d_{|}^{0};d_{-}^{0};d_{|}^{1};d_{-}^{1};d_{|}^{2};d_{-}^{2};\Sigma_{5}^{0}\}. \tag{14}$$

4.5.3 *Training image set*

For the training stage, keeping in mind the monoscale experience, we continue to use a synthetic image set. But in order to take into account the noisy cases, we have studied the training conditions for them. The simplest way to cope with such cases is to put them in the training set of images. In this approach, we have noticed that it is only when the standard deviation of the noise in the image to be segmented is greater than the contour amplitude that noisy training images need to be used. And the noise added to the previously presented synthetic universal training image must be about half the noisy standard deviation expected in the image to segment. But this training scheme has to be performed carefully, and it must be reserved for heavily noised images, because it leads to discrepancies in the classifier performance for clean images. Therefore, in the standard case, we choose to use the synthetic universal training image defined in the monoscale approach.

4.5.4 *Some results*

The results obtained in this multiscale scheme are presented in Figs. 14, 15, and 16. These examples clearly show the complementarity between the information given by the various scales for the segmentation. For the noised case

Fig. 14 Results on noisy (Gaussian noise of standard deviation 22) real images for identical ND and P. From left to right: segmented image by thresholding (upper threshold=17), segmented image by merging system. These two images are represented by approximately the same point on the curves of Fig. 13: ND≈27, P≈70. The third image results from the merging of the image "Woman." (See original image "Woman" in Fig. 16.)

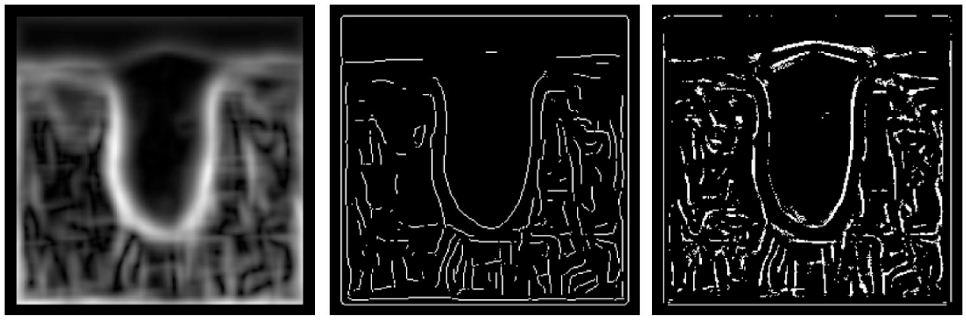


Fig. 15 Result on a gradient image of a soldering area (presenting closed and blurred edges). From left to right: original image; image segmented by hysteresis; merged image. The results here are very encouraging, the objective being to detect the shape of the soldering zone shown on this image.

(Fig. 14) the multiscale approach gives robustness, and only the edges that persist across the scales are detected, the localization being given by the finest scale.

We have also verified that for good quality images (no blur, no noise), the performance of the multiscale scheme is the same as obtained with the monoscale approach. We have quantized the performance of this algorithm in the same way as before, by using the (P,ND) graph (Fig. 13) and the Pratt figure of merit (Fig. 12). The latter curves (Fig. 12) show that the multiresolution method is only half as sensitive to the setting of the training parameters setting as the classical method (hysteresis threshold). In Fig. 13 we have another comparison between the two methods, applied to real images instead of the synthetic Pratt test image.

The examples in Figs. 14, 15, and 16 show clearly the advantages of our method for noisy and blurred images.

5 Conclusion

We have presented a multiscale edge detection algorithm whose aim is to detect edges of any slope. Our work is based on a generalization of the Canny-Deriche filter, characterized by a more realistic edge than the traditional step-shape. The filter impulse response has been used to generate multiscale edge detection. The filter has been implemented in an ASIC. This circuit is able to estimate three levels of decomposition in real time (40 ms) for a 512×512 image. For the merging of the edge information, a geometrical classifier already developed in our laboratory has been used. The merging system has nonoptimal performance from the classifier point of view, but it is a real-time (video-rate) nonparametric system. The segmentation system thus set up, after the training phase, does not require any adjustment or parameter setting. The main original property of this algorithm is that it leads to a binary edge image without any threshold setting.

The system is robust to illumination variation. Moreover, in case of high deterioration of the image by noise, the merging remains very robust, since the edge points obtained are still localized near to the real edges. On the contrary, the classical method gives in this case a heavily noised result. If the degradations are constant in the image (stationary noise, constant width of the blurred edges), the classical method could lead to the best results on choosing the right scale and the correct thresholds.

6 Appendix: Recursive Implementation

Let x be the signal image, y the response. For the derivative filter, the Z transform leads us to the following difference equations:

$$y_i^+ = N(a_1 x_{i-1} + a_2 x_{i-2}) - b_1 y_{i-1}^+ - b_2 y_{i-2}^+ - b_3 y_{i-3}^+ , \tag{15}$$

$$y_i^- = -N(a_1 x_{i-1} + a_2 x_{i-2}) - b_1 y_{i-1}^- - b_2 y_{i-2}^- - b_3 y_{i-3}^- , \tag{16}$$

with

$$a_1 = (ksT+1)\exp(-msT) - \exp(-sT),$$

$$a_2 = -(ksT-1)\exp[-(m+1)sT] - \exp(-2msT)$$

Fig. 16 Nondeteriorated image. From left to right: original approach; image segmented by hysteresis; merged image.

and

$$b_1 = -2\exp(-msT) - \exp(-sT),$$

$$b_2 = \exp(-2msT) + 2\exp[-(m+1)sT],$$

$$b_3 = -\exp[-(2m+1)sT],$$

and with

$$\frac{N}{|\Sigma_{i=1}^{+\infty}\psi(i)|} = 1. \tag{17}$$

Here T represents the sampling period, which is chosen as equal to 1 as in previous work. The total response is the sum of the two responses obtained by two recursive relations of third order:

$$y(i) = y^+(i) + y^-(i), \tag{18}$$

$$y_{p,q}(i) = y(2^{-p}i - q). \tag{19}$$

Equation (19) gives the multiscale and sampled response. In the same way the recursive response for the smoothing is obtained:

$$y_i^+ = M(c_0^+ x_i + c_1^+ x_{i-1} + c_2^+ x_{i-2}) - d_1 y_{i-1}^+ - d_2 y_{i-2}^+ - d_3 y_{i-3}^+, \tag{20}$$

$$y_i^- = M(c_0^- x_i + c_1^- x_{i-1} + c_2^- x_{i-2} + c_3^- x_{i-3} - d_1 y_{i-1}^- - d_2 y_{i-2}^- - d_3 y_{i-3}^-, \tag{21}$$

with

$$c_0^+ = m(m-1) - k,$$

$$c_1^+ = (-kms + k + m - 2m^2)\exp(-msT) + (k+m) \times \exp(-sT),$$

$$c_2^+ = (kms - k - m)\exp[-(m+1)sT] + m^2\exp(-2msT),$$

$$c_0^- = c_0^+ - \phi(0),$$

$$c_1^- = c_1^+ - d_1\phi(0),$$

$$c_2^- = c_2^+ - d_2\phi(0),$$

$$c_3^- = -d_3\phi(0),$$

$$\phi(0) = m(m-1) - k$$

and

$$d_1 = -2\exp(-msT) - \exp(-sT),$$

$$d_2 = \exp(-2msT) + 2\exp[-(m+1)sT],$$

$$d_3 = -\exp[-(2m+1)sT],$$

and with

$$\frac{M}{|\Sigma_{i=-\infty}^{+\infty}\phi(i)|} = 1. \tag{22}$$

Acknowledgment

The authors wish to thank Prof. E. Bourenanne for his invaluable help in the design of the algorithms used in this work.

References

1. J. W. Modestino and R. W. Fries, "Edge detection in noisy images using recursive digital filter," *Comput. Graph. Image Process.* **1(6)**, 409–433 (1977).
2. K. S. Shanmugam and J. A. Green, "An optimal frequency domain filter for edge detection in digital pictures," *IEEE Trans. Pattern Anal. Mach. Intell.* **1**, 37–49 (1979).
3. W. H. H. J. Lunscher, "The asymptotic optimal frequency domain filter for edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **5(6)**, 678–679 (1983).
4. J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **8(6)**, (1986).
5. D. Demigny and T. Kamleh, "A discrete expression of Canny's criteria for step edge detection performances evaluation," *IEEE Trans. Pattern Anal. Mach. Intell.* **19(11)**, 1199–1211 (1997).
6. D. Demigny, "On optimal linear filtering for edge detection," *IEEE Trans. Image Process.* **11(7)**, 728–737 (2002).
7. A. P. Witkin, "Scale-space filtering," in *Proc. Int. Joint Conf. on Artificial Intelligence*, Alan Bundy, Ed., pp. 1019–1022 (1983).
8. D. J. Williams and M. Shah, "Multiple scale edge linking," in *Applications of Artificial Intelligence*, *Proc. SPIE* **1095**, 13–24 (1989).
9. W. F. Bischof and T. Caelli, "Parsing scale-space and spatial stability analysis," *Comput. Vision Graphics Image Process.* **42**, 192–205 (1988).
10. F. Bergholm, "Edge focusing," *IEEE Trans. Pattern Anal. Mach. Intell.* **9(6)**, 726–741 (1987).
11. Y. Lu and R. C. Jain, "Reasoning about edges in scale space," *IEEE Trans. Pattern Anal. Mach. Intell.* **14(4)**, 450–468 (1992).
12. M. Forthoffer, J. P. Girod, and J. Bremont, "Edge detection using wavelet transform and neural networks," *Trait. Signal* **8(6)**, 431–440 (1991).
13. E. Bourennane, M. Paindavoine, and F. Truchetet, "An improvement of Canny-Deriche filter for ramp edge detection," *Trait. Signal* **10(4)**, 297–310 (1993).
14. M. Petrou and J. Kittler, "Optimal edge detectors for ramp edges," *IEEE Trans. Pattern Anal. Mach. Intell.* **13(5)**, 1483–1491 (1991).
15. R. Deriche, "Using Canny's criteria to derive a recursively implemented optimal edge detector," *Int. J. Comput. Vis.* **1(2)**, 167–187 (1987).
16. E. Bourennane, P. Gouton, M. Paindavoine, and F. Truchetet, "Generalization of Canny-Deriche filter for detection of noisy exponential edge," *Signal Process.* **82(10)**, 1317–1328 (2002).
17. F. Truchetet, O. Laligant, J. Mitéran, and E. Bourennane, "Frames of wavelets for edge detection," in *Wavelet Applications in Signal and Image Processing*, *Proc. SPIE* **2303**, 141–152 (1994).
18. Sarifuddin, "Implantation de la transformée en ondelettes pour la détection de contour," Thèse, Univ. de Bourgogne (1995).
19. J. Mitéran, P. Gorria, and M. Robert, "ASIC de segmentation d'images en temps réel par apprentissage," in *Proc. 14 Collog GRETSI*, pp. 1071–1073 (1993).
20. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, pp. 230–243, Wiley, New York (1973).
21. M. Ichino, "A nonparametric multiclass pattern classifier," *IEEE Trans. Syst. Man Cybern.* **9(6)**, 345–352 (1979).
22. J. Mitéran, P. Gorria, and M. Robert. "Parallel processor for image segmentation," in *IEEE Conf. MELECON 94*, pp. 12–15 (1994).
23. G. Mattews and J. Hearne, "Clustering without a metric," *IEEE Trans. Pattern Anal. Mach. Intell.* **13(2)**, 175–184 (1991).
24. S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Trans. Pattern Anal. Mach. Intell.* **14(7)**, 710–732 (1992).

Olivier Laligant received his PhD degree in 1995 from the Université de Bourgogne, France. He is an assistant professor in the Laboratory of Electronic, Computing, and Imaging Sciences (Le2i) there. His research interests are focused on multiscale edge detection, merging of data, wavelet transforms, and—recently—motion estimation.

Frédéric Truchetet received the master's degree in physics at Dijon University, France, in 1973, and a PhD in electronics at the same university in 1977. He was with Thomson-CSF for two years as a research engineer, and he is currently a full professor and the head of the image processing group at Le2i, Laboratory of Electronic, Computer, and Imaging Sciences, at Université de Bourgogne, France. His research interests are focused on image processing for artificial vision inspection and particularly on wavelet transforms, multiresolution edge detection, and image compression. He is member of SPIE, IEEE, and ISIS (a research group of CNRS).

Johel Miteran received the PhD degree in image processing from the University of Burgundy, Dijon, France, in 1995. Since 1996, he has been an assistant professor at Le2i, University of Burgundy. He is now engaged in research on classification algorithms, face recognition and access control problems, and real-time implantation of these algorithms on software and hardware architectures.

Patrick Gorria received the degree of engineer from the ENSAM, France, in 1980. From 1980 to 1982, he was in Mexico. He received the PhD degree from the University of Burgundy, France, in 1984. Since 1993, he has been a professor at the Laboratory Le2i of Image Processing in the Institut Universitaire de Technologie (IUT), Le Creusot, where he is engaged in research and teaching on quality control by artificial vision. He has contributed to several areas of classification, sensors, and other systems and applications using artificial vision systems for quality control. He is presently a principal researcher at Le2i, where he conducts research and development of defect detection for quality control by artificial vision.