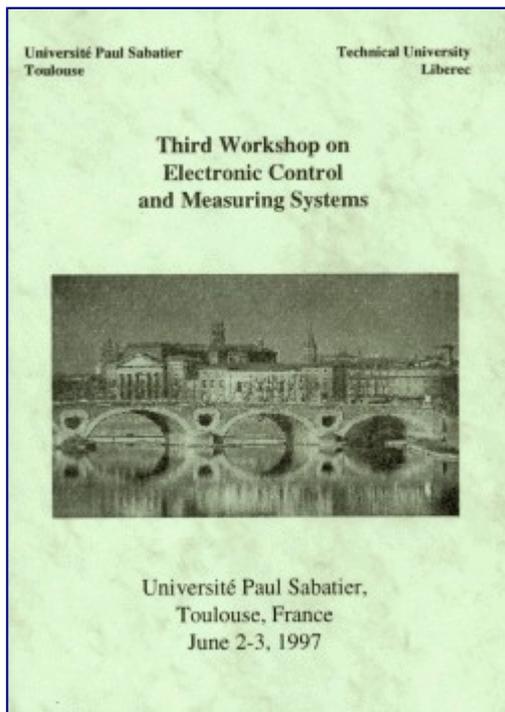


Edge and line detection in low level analysis

*In this paper, we describe a new straight line detection algorithm, using derivative-based edge operators, which works without internal thresholding nor any internal parameterization (see below). It has been presented in the **Third Workshop on Electronic Control and Measuring Systems**, Université Paul Sabatier (Toulouse), on June 2-3, 1997. Although minor modifications have occurred, the version presented here is a copy of the version published in the proceedings, a French version is available.*



- [Authors](#)
- [1. Summary](#)
- [2. Line detection method](#)
- [3. Results and application](#)
- [Bibliography](#)

Authors

Baptiste Marcel and **Michel Cattoen**
Groupe Signaux, Images et Communication,

Laboratoire Électronique
École Nationale Supérieure d'Électrotechnique, Électronique et Hydraulique de Toulouse,
2, rue Charles Camichel
BP7122
31071 Toulouse CEDEX 7
France

1. Summary

Edge detection and line extraction are important and critical components in an image understanding system, the result of the extraction being the basis of the high level processes.

In this paper, we describe a new straight line detection algorithm, using derivative-based edge operators, which works without internal thresholding nor any internal parameterization. This property delays the detection decisions up to the high levels of the image analysis system rather than in the low level line detection processes. This reduces the amount of parameters to adjust in the supervising image understanding system. This method also allows low contrast lines to be detected, without an excess of false alarms, because the spurious isolated noise pixels marked as edges in the first phases will not be validated in the following phases. The method is mostly based upon simple filter computations which should be able to produce fast results in hardware embodiment.

2. Line detection method

A line is a connected group of aligned edge pixels (referred as *edgels*), and a corner is a group of edgels that form together two jointive non-parallel lines. We will focus in this paper on corner representing angles around 90°.

Line-detection algorithms based on contour extraction usually involve the following stages of image processing: smoothing, edge detection, edge thinning, edge linking, chain straightening and line correction. In our algorithm, the second stage includes two derivative processes using well-known laplacian and gradient filters [WANG94] to determine edge location and orientation. The three last stages process the lines extraction. The above-mentioned smoothing and thinning stages known from literature are not processed as they are implicit in the algorithm by the choice of the derivative method.

The algorithm is thus executed in 5 phases : edge detection, edge direction computation, edge linking, chain straightening and line restoration.

2.1 Edge detection

In phase 1, thin edges are detected on zero crossings of the laplacian convolution filter's response. The filter used is the sum of the second derivative one-dimensional filters in either 2 or 4 directions (using diagonals). The size of the laplacian filter determines the bandwidth of the implicit smoothing in the filter process.

The White-Rohrer (two directions, [WHIT83]) and Marr-Hildreth (four directions, [HARA84]) filters use the matrixes M_{WR} and M_{MH} (equations 1 and 2).

$$M_{WR} = \begin{pmatrix} & & 1 & & \\ & & 0 & & \\ 1 & 0 & -4 & 0 & 1 \\ & & 0 & & \\ & & 1 & & \end{pmatrix} \quad (1), \quad M_{MH} = \begin{pmatrix} \frac{1}{\sqrt{2}} & & & & \\ & 1 & & & \frac{1}{\sqrt{2}} \\ 1 & -4 & -2\sqrt{2} & & 1 \\ & & & & \\ \frac{1}{\sqrt{2}} & & & & \frac{1}{\sqrt{2}} \end{pmatrix} \quad (2)$$

We address g the image to scan. g is a function of the plane defined on a grid which each node is identified by two coordinates (i,j) ; the derivations of the convolution matrixes are developed in equations 3 and 4.

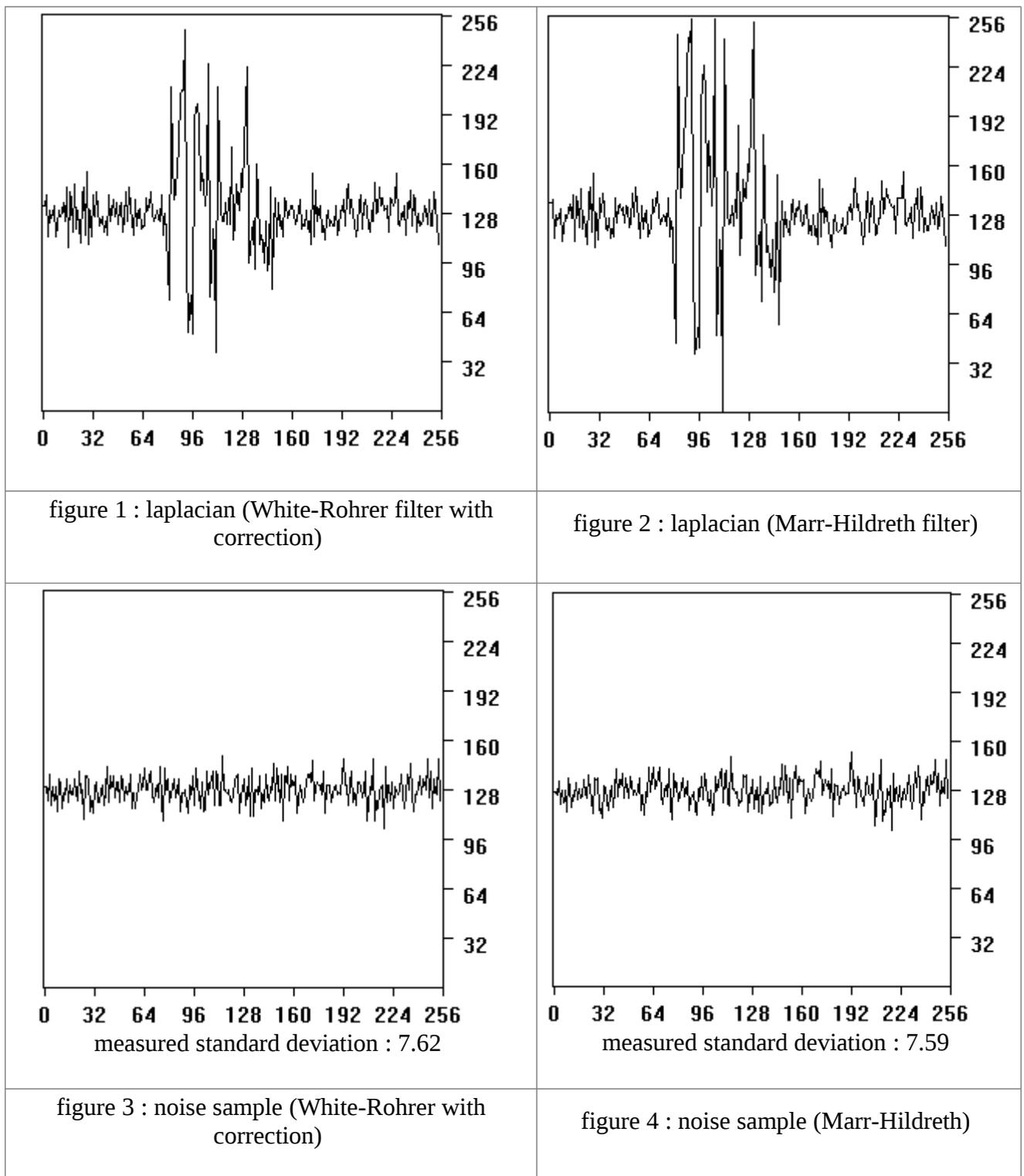
$$M_{WR} : \frac{\partial^2 g(x,y)}{\partial x^2} = g(i+2,j) + g(i-2,j) + g(i,j+2) + g(i,j-2) - 4g(i,j) \quad (3)$$

$$M_{MH} : \frac{\hat{g}^2(i, j)}{\hat{\sigma}^2} = g(i+1, j) + g(i-1, j) + g(i, j+1) + g(i, j-1) - 4g(i, j) \quad (4)$$

$$+ \frac{1}{\sqrt{2}} (g(i+1, j+1) + g(i-1, j+1) + g(i+1, j-1) + g(i-1, j-1) - 4g(i, j))$$

The last one (four directions) presents a better signal-to-noise ratio (as can be seen on figures 1 to 4), but a longer processing duration, as the diagonal intensity of the zero-crossing must be corrected by the $1/\sqrt{2}$ factor and thus needs floating point calculations. However, a quick comparison between equations 3 and 4 indicates that the Marr-Hildreth computation can be performed with only twice the computation effort of the White-Rohrer one.

The figures 1 and 2 represent the response on White-Rohrer (fig. 1) and Marr-Hildreth (fig. 2) filter in a zone where edgels were present ; the figures 3 and 4 represent the same filters responses in a no-edgels zone. The intensities have been corrected to present the same noise levels in both filters (the noise measurement is the standard deviation of the no-edgels samples). Comparing intensities of laplacian response on edgel zone versus average response on non-edgel zone, taking the correction into account, the signal-to-noise ratio is better on the Marr-Hildreth filter than on the White-Rohrer one.



A pixel is marked as an edge if the laplacian's response presents a zero-crossing in either line or column direction. The intensity of the edge is credited as the highest intensity of horizontal and vertical zero-crossings, without thresholding.

Zero-crossings of the laplacian represent inflection points in the direction of the gradient of the image (which is a \mathbf{R}^2 to \mathbf{R} function). However, some inflection points of the image do not represent edges. This occurs ([FLEC92]) where the sign of the first derivative is the same as the sign of the third derivative. This may be demonstrated on examples on \mathbf{R} to \mathbf{R} functions that will be continuously derivable in the neighbourhood of the considered point.

To be declared as edge, an inflection point requires the following conditions :

- If f is increasing, the inflection point is on a local maximum of f' , which means: f' positive, f'' null and decreasing and f''' negative (figure 5.a)

- If f is decreasing, the inflection point is on a local minimum of f , which means: f' negative, f'' null and increasing and f''' positive (figure 5.c)

In these two cases, the sign of f' is different from the sign of f''' . In the other two cases (fig. 5.b and 5.d), the inflection point does not belong to an edge.

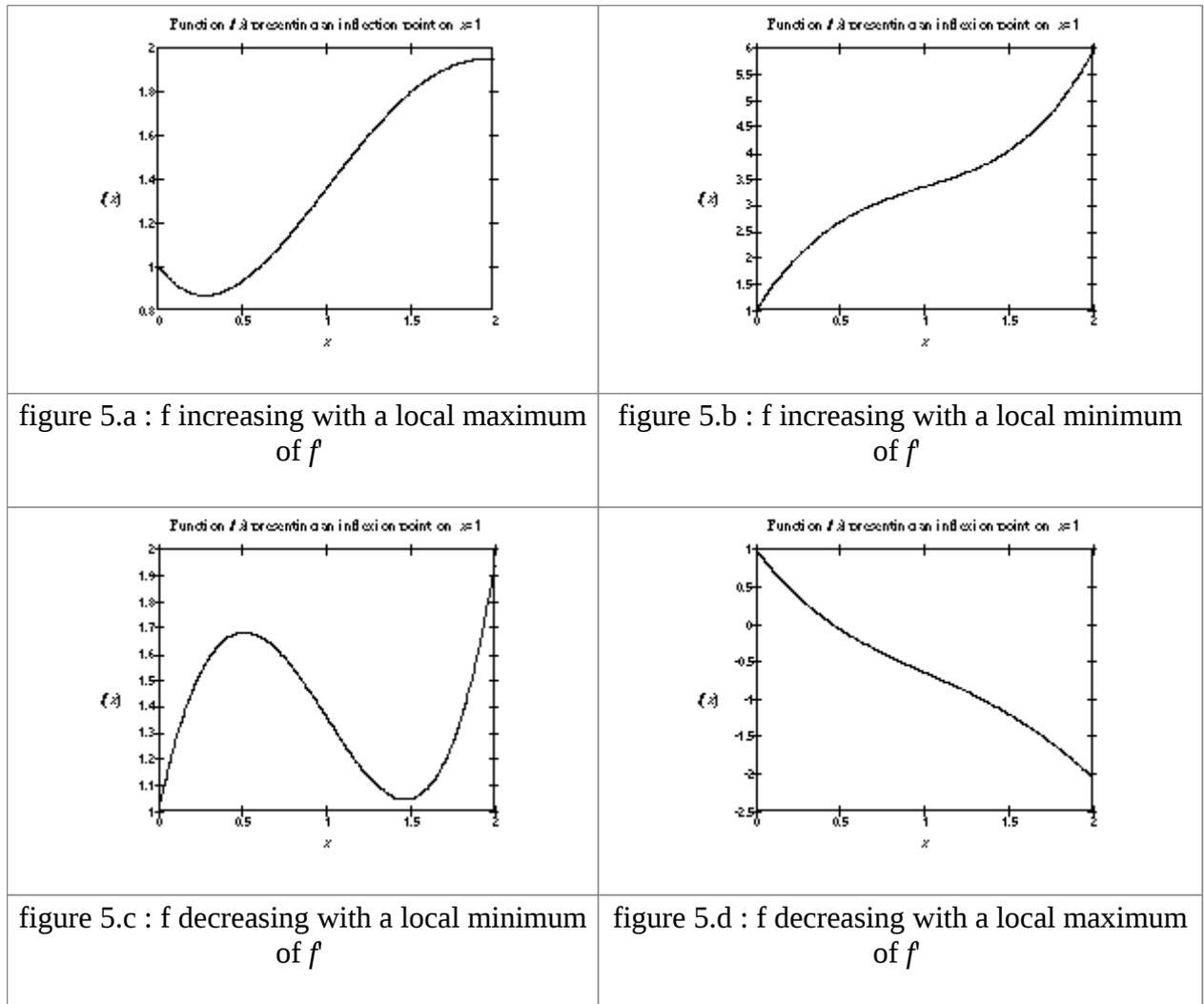


figure 5 : Four kinds of inflection points on $x=1$ (edges on 5.a and 5.c)

The pixels in the cases 5.b and 5.d are erroneously marked as edge in our method. The identification of these non-edge inflection points would invoke a third derivation process (f'''), but it appeared not to be necessary, as these false alarm pixels disappear in the next phases.

The usual thinning phase of chain extraction algorithms is not needed here because the zero-crossing of the laplacian delivers thin edges. However, neighbouring edges in gradient direction may appear [\[TORR86\]](#) when anti-parallel boundaries are one pixel apart, or when the algorithm associates vertical and horizontal zero-crossing edges around a corner.

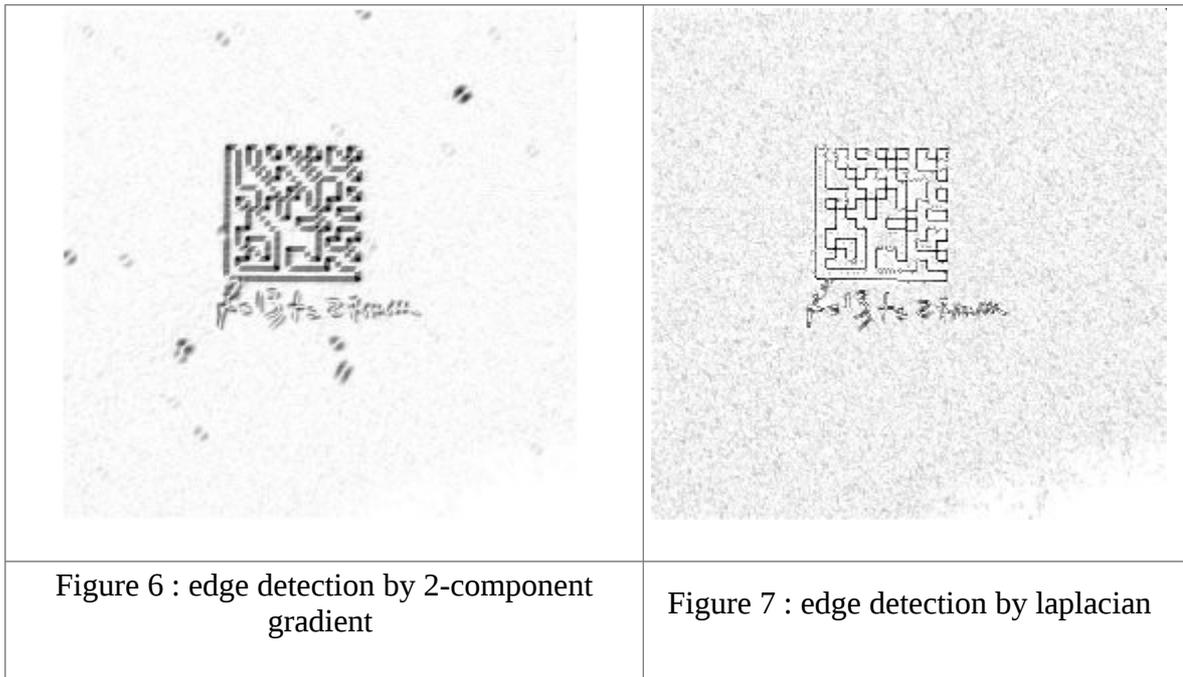
2.2 Edge direction computation

In phase 2, the orientation of each detected edge is computed by 2 directional gradient filters (vertical and horizontal) which estimate the 2 coordinates of the gradient vector. The matrixes used and the derivation operations are shown on equations 5 et 6.

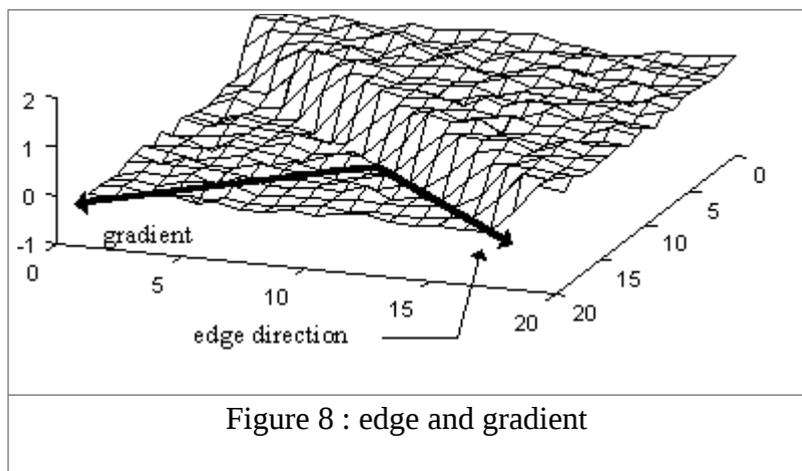
$$\text{Vertical component : } \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}, \frac{\partial g(i, j)}{\partial x} = g(i, j+1) - g(i, j) \quad (5),$$

Horizontal component : $(1 \ -1 \ 0)$, $\frac{\partial g(x,y)}{\partial x} = g(i-1,j) - g(i,j)$ (6)

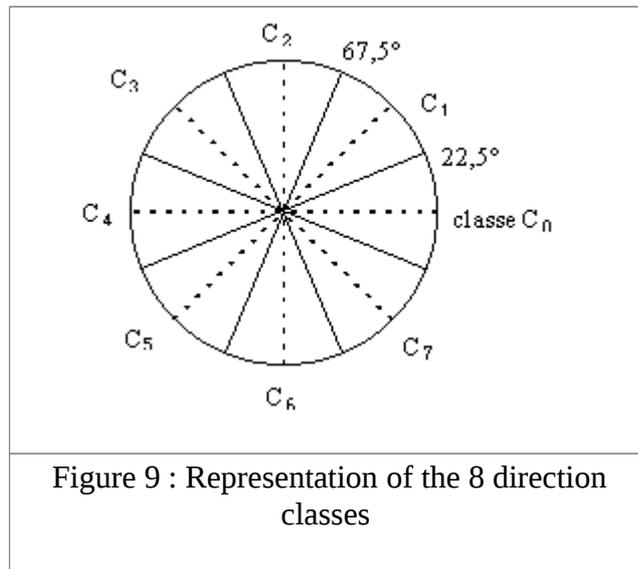
These operators are not very accurate for localisation purposes. One can notice it on figures 6 and 7 which represent responses from edge detection by these filters (fig. 6) and the White-Rohrer laplacian (fig. 7). These are thus not very efficient for edge detection.



The figure 8 represents the relation between an edge and a gradient in the image.



The gradient direction will not need to be known with a great precision, as it is only used to resolve 8-pixel neighbouring in the link process in the next phase. The gradient direction is represented with one of the 8 possible directions from the central pixel ; this leads to a 3 bits storage, and a maximum error of 22.5° (figure 9).



2.3 Edge linking

In phase 3, the edges are linked into chains according to their proximity and gradient direction. In this step, an edge is linked with its neighbour if the line joining the two edges and the gradient direction are no more than 45 degrees apart. The result is a set of chains representing thin edges (1 pixel wide).

2.3.1 Linking process

Figures 10 and 11 present an example of the chain process : a zoom on an image around a corner (fig. 10), and the directions and intensities of its edges (fig. 11).

Whereas an edge occurs between two pixels, the edge is set on the pixel that presents the weakest intensity (the darkest). The null-intensity edges are not represented. The intensity and direction is shown for every edge. The length of the arrow is proportional to the intensity of the edge. The direction of the edge has been represented with dotted lines and fixed length for some pixels.



Figure 10 : zoom on a corner in an image

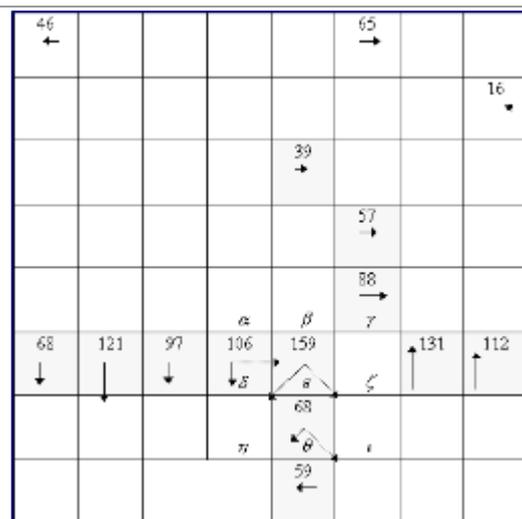


Figure 11 : directions and intensities of gradient in the same zone

The edge direction on pixel *delta* can not allow linking with pixels other than *beta*, *epsilon* or *theta*. The closest one (*epsilon*) presents an edge which direction cannot allow linking with other than *beta*, *alpha* or *delta*. The *delta-epsilon* link is thus compatible with both gradient directions. Furthermore, both the edge

directions of pixels δ and ϵ belong to neighbouring classes (0 and 7 modulo 8). δ to ϵ can thus be linked in this order.

The next pixel will be θ because this one is one of the three eligible successors of ϵ (which are ζ , ι et θ), ϵ is one of the three eligible predecessors of θ (which are ϵ , δ and η), and the edge directions of pixels ϵ and θ are the same.

2.3.2 Non-maximum edge suppression

This chaining is not relevant in case of neighbouring edges in gradient direction which indicates aggregated anti-parallel boundaries in high frequency textured pictures. The edge map needs a non-maximum edge suppression before phase 3 is processed in order to reduce the image's texture frequency and sample the edges to deliver to the next phase. The non-maximum suppression of edges removes the edgels which present an edgel of weakest intensity next to them (in a direction orthogonal to their own direction). For each edgel pe , the edge intensities of its side neighbours (in the direction of its gradient) is computed. If one of these two neighbours has its intensity greater than pe 's intensity, then pe is removed from the edge map. Actually, few edgels will be removed by this process.

An example of the non-maximum edge suppression process is presented on figures 12 and 13: a zoom on an image around a corner (fig. 12), and the directions and intensities of its edges (fig. 13). The corner edgel (of intensity 164) has a neighbour in its gradient direction, which presents an edge of highest intensity (204). The corner edgel will be removed.

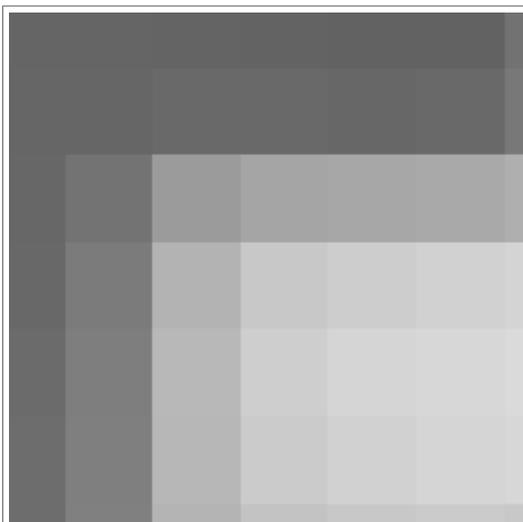


Figure 12 : zoom on a corner in an image

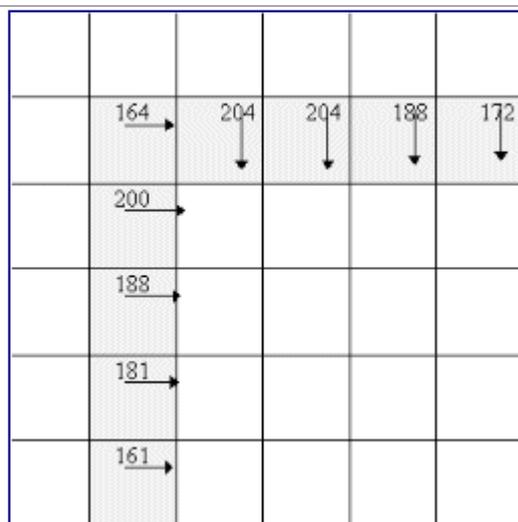


Figure 13 : directions and intensities of gradient in the same zone

2.3.3 Short chain elimination

As the zero-crossing response of the previous step is not thresholded, the signal before this step presents many spurious edges due to local weak intensity discontinuities (commonly referred as *noise*) and not-edge inflection points. As these false alarm edges are more or less randomly distributed, they usually do not aggregate in chains bigger than 3 pixels. Elimination of short chains, i.e. smaller than 3 or 4 pixels, removes from the response all the noise which had been detected by the first phase and preserves the low contrast lines.

2.4 Chain straightening

In phase 4, the chains are split to obtain straight lines. The algorithm breaks each chain at the furthest pixel from its chord until its distance reach a value smaller than 1 or 2 (figure 14). As lines may appear as

low curvature segments in geometrically distorted pictures, their detection may imply a higher threshold for chain breaking depending on the vision system used. This method forces each pixel to belong to a unique line, which excludes jointive or secant lines.

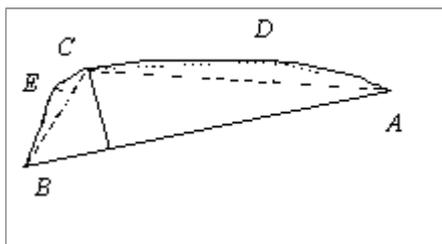


Figure 14 : straightening of a chain

2.5 Line restoration

In phase 5, lines' extremities are shifted to recover altered corners, and separated lines are merged to recover isolated missing edge elements.

Some lines produced by phase 4 somehow misfit the actual lines of the image. The main cause is corner splitting. Since thin edges are used, an edgel cannot have more than two neighbours. Only edgels which present continuous variation in their direction are linked together. For these two reasons, an edge chain cannot contain an angle greater than 45° . It may also appear that a corner is split due to weak contrast or high frequency noise in the picture.

The corner dislocation does not interfere on a critical basis with our purpose which focus on images that present a bi-level intensity distribution, based upon binary images in the significant area. In this kind of images, lines intersections cannot occur. The restoration phase used here is sufficient for our aim.

Figures 16 and 17 present the lines extracted from figure 15 with and without lines restoration. The set of line is more consistent after the restoration phase, even if some defects are not actually visible. On figure 16, the lines on the left of the Data-Matrix symbol (in the centre of the image) are shown as two joint orthogonal lines. They are actually separated, and have their extremities lying on neighbouring pixel (of exactly one pixel apart) which appear as one consistent corner. The line which is indicated by a light arrow in picture 17 appears to be broken in the picture 16 and is corrected by the restoration phase.



Figure 15 : image containing straight lines



Figure 16 : raw line extraction from figure 15

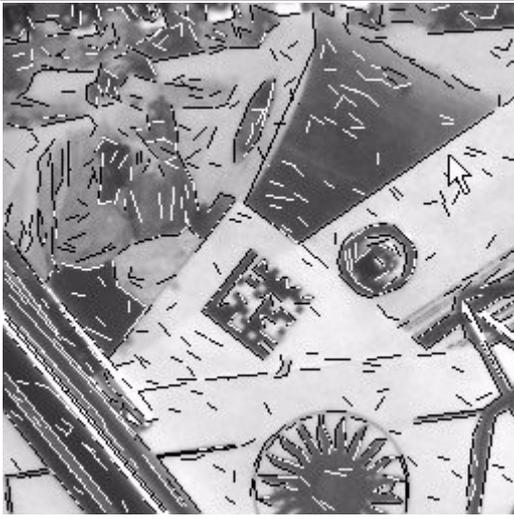


Figure 17 : restored lines from figure 16 (superimposed on original image)

3 Results and applications

The described algorithm works well in the scope of applications for which it was designed. The constraints and properties of these applications are presented here, they will lead to the reason for which we developed this method.

The method has been designed for the detection process of bi-level bar-codes derivatives in pictures. (Data-Matrix codes are shown on figures 15 and 18). These symbol are printed on a bi-level basis, and attained by way of camera in real-world situation, which include optic and projective distortion, unreliable lighting... The aim of the process is to detect and localise the symbol.

The method delivers multiple little linked lines as response from a curvature boundary, which results in the polygonization of the objects' boundary ; if this may be a handicap for some image understanding processes, it does not affect our objective. Indeed, the symbol is composed of shapes with straight borders ; the shapes and texture characteristics of areas outside the symbol are of no interest, and the way that our line detector reacts on these textures does not concern us for this application as long as it cannot be misinterpreted as symbol characteristics (which are long lines).

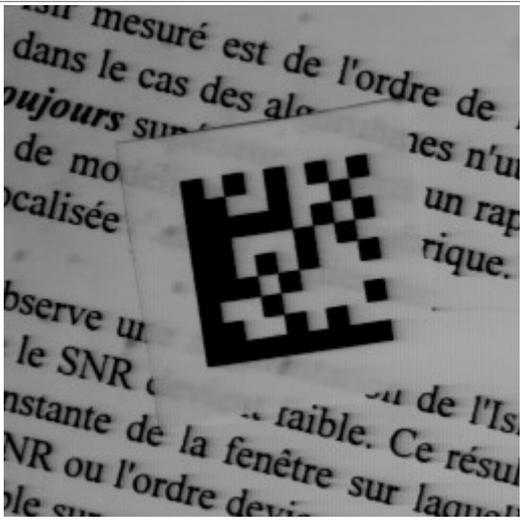
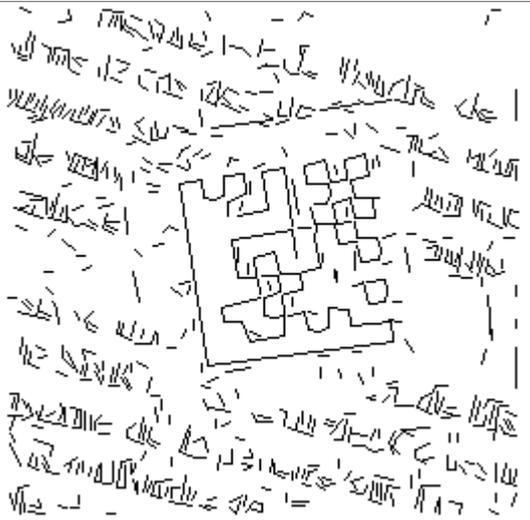
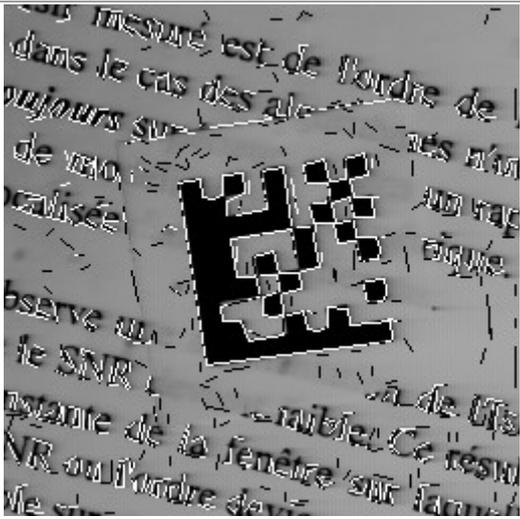
The symbol is itself a bi-level pattern, which means that it can be represented unambiguously by a set of shapes, or by a set of closed thin boundaries representing these shapes. This characteristic implies that the original picture, in the area where the symbol stands, does not contain no-edge inflection points, nor wide edges with uncertain localisation, nor vertex or line extremities where chain linking could not be solved unambiguously. The derivation process used (zero-crossing of the laplacian which naturally produces thin closed boundaries) as well as the non-maximum edge suppression and the iterative linking process may produce spurious or missing edge element, but these hindrances barely occurs in our application field.

As we have explained in § 2.5, our methods induces corner dislocation which is corrected by the restoration phase. As symbol images do not contain lines intersecting, disjoint lines on corner of symbol shapes are properly recovered by the restoration phases.

These detectors are to be implemented in human-usable devices which need fast calculation. As several of this method's phases are based upon linear filter operators, fast processing of the low level phases should be achieved using proper hardware implementation.

However, the method does not work well on little lines (2 to 3 pixels). These lines cannot be properly constructed out of responses from filters of size 2×2 or 3×3 , and are irrevocably lost as edge-location errors induce some significant errors in the line directions. As a consequence, the short line elimination of phase 3 does not depreciate the final detection results as short lines were unreliably detected anyway.

Figures 18 to 20 present the result of the detection process on Data Matrix images.

	
<p>Figure 18 : image of a Data-Matrix symbol</p>	<p>Figure 19 : line detection result</p>
	
<p>Figure 20 : line detection result superimposed on image 18</p>	

The lines detected here are sufficient to detect the symbol. The spurious little lines that may occur cannot lead to a misdetection. Faint edge lines, such as the limit of the quiet zone at the right and top of the symbol, are detected although these were not detected as continuous long lines.

Bibliography

- [WANG94] Mao-Jiun J. Wang, Shiau-Chyi Chang, Chih-Minh Liu and Wen-Yen Wu, « A new edge detection method through template matching », *International journal of Pattern Recognition and artificial intelligence*, Vol. 8, n°4, 1994, p. 899-917.

- [WHIT83] J. M. White and G. D. Rohrer, « Image thresholding for optical character recognition and other applications requiring character image extraction », *IBM J. res develop.*, Vol. 27, n°4, July 1983, p. 400-411.
- [HARA84] Robert. M. Haralick, « Digital step edges from zero crossing of second directional derivatives », *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 6, n°1, January 1984, p. 58-68.
- [FLEC92] Margaret M. Fleck, « Some defects in finite-difference edge finders », *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 14, n°3, March 1992, p. 337-345.
- [TORR86] Vincent Torre and Tomaso A. Poggio, « On edge detection », *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 8, n°2, March 1986, p. 147-163.
-

LUNDI 2 JUIN 1997

9h : OUVERTURE DU CONGRES « E.C.M.S. 97 »

G. LARROUY, Président U. P. S.

Z. KOVAR, Vice-Recteur T. U. LIBEREC

THEME : "CONTROL"

Chairman J. C. MUGARZA (Spain)

**9h30 REAL TIME ADAPTATIVE CONTROL OF A RAPID THERMAL
PROCESSING SYSTEM**

B. DAHHOU

G. ROUX

S. MORALES

10h UNIVERSAL METHODS OF DYNAMICAL OPTIMIZATION

V. ZADA

**10h30 CONTROLLER WITH A FLEXIBLE FEEDBACK FOR HIGH ORDER
CONTROLLER SYSTEMS OR SYSTEMS WITH DELAY**

B. HANUS

M. HERNYCH

R. SINDELAR

Pause et visite des stands d'exposition.

11h30 THE CONTROL SYSTEM FOR SMALL FLUIDIZED-BED BOILERS

B. JANEBA

B. JANECEK

J. CYRUS

J. VALHA

P. HEJZLAR

F. HRDLICKA

V. MICHNA

L. TUMA

J. LUKES

**12h BASED ON PLDS PROGRAMMABLE LOGIC CONTROLLER WITH
REMOTE I/O GROUPS**

E. HRYNKIEWICZ

THEME : "PETRI NETS"

Chairman B. DAHHOU (France)

14h PETRI NETS FOR MANUFACTURING SYSTEMS : FROM SPECIFICATION TO REAL TIME CONTROL

J. C. MUGARZA CORTABARRIA

J. R. TORREALDAY APRAIZ

14h30 CONTROL OF THE AUTOMATED SPINNING MILL

P. RYDLO

Pause et visite des stands d'exposition.

THEME : "SIGNAL AND SPEECH PROCESSING"

Chairman J. NOUZA (Czech Republic)

15h30 NON-LINEAR MODELING OF ACOUSTIC PULSE SIGNALS BY WAVELET NETWORKS

S. NADAUD

J. F. TROUILHET

16h A CDHMM GENERATOR AND ITS USE IN SPEECH PROCESSING

J. NOUZA

D. HAJEK

16h30 FUNDAMENTAL FREQUENCY CONTROL IN CZECH TEXT TO-SPEECH SYNTHESIS

J. TUCKOVA

P. HORAK

17h MULTIMODAL INTERFACE FOR ROBOTIZED APPLICATION

N. VIGOUROUX

M. POTIER

P. BAILLION

MARDI 3 JUIN 1997

THEME : « IMAGE PROCESSING »

Chairman M. KOSEK (Czech Republic)

9h EDGE AND LINE DETECTION IN LOW LEVEL ANALYSIS

M. CATTOEN

B. MARCEL

9h30 COLOR SEGMENTATION IN PRINCIPAL REGIONS FOR NATURAL OUTDOOR SCENES

R. MURRIETA CID

M. BRIOT

P. LASSERRE

10h MACHINE LEARNING FOR IMAGE ANALYSIS

L. MASCARILLA

J. DESACHY

Pause et visite des stands d'exposition.

THEME « INSTRUMENTATION AND TEST »

Chairman J. NOSEK (Czech Republic)

11h INSTRUMENTATION FLEXIBLE : SPECIFICATION BASEE SUR UNE ARCHITECTURE DE COMMUNICATION RECONFIGURABLE

B. ARAGON

A. KARIMI-FARD

D. ROUGE

11h30 CONCEPTION OF AN INSTRUMENTATION : A MIXED ASIC STUDY FOR MULTI-WAY ACQUISITION

A. KARIMI-FARD

P. LEMAN

B. ARAGON

THEME : "PETRI NETS"

Chairman B. DAHHOU (France)

14h PETRI NETS FOR MANUFACTURING SYSTEMS : FROM SPECIFICATION TO REAL TIME CONTROL

J. C. MUGARZA CORTABARRIA

J. R. TORREALDAY APRAIZ

14h30 CONTROL OF THE AUTOMATED SPINNING MILL

P. RYDLO

Pause et visite des stands d'exposition.

THEME : "SIGNAL AND SPEECH PROCESSING"

Chairman J. NOUZA (Czech Republic)

15h30 NON-LINEAR MODELING OF ACOUSTIC PULSE SIGNALS BY WAVELET NETWORKS

S. NADAUD

J. F. TROUILHET

16h A CDHMM GENERATOR AND ITS USE IN SPEECH PROCESSING

J. NOUZA

D. HAJEK

16h30 FUNDAMENTAL FREQUENCY CONTROL IN CZECH TEXT TO-SPEECH SYNTHESIS

J. TUCKOVA

P. HORAK

17h MULTIMODAL INTERFACE FOR ROBOTIZED APPLICATION

N. VIGOUROUX

M. POTIER

P. BAILLION