On corner and vertex detection

Gerard Giraudon, Rachid Deriche

INRIA

UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.: (1) 39 63 55 11

Rapports de Recherche

N° 1439

Programme 4
Robotique, Image et Vision

ON CORNER AND VERTEX DETECTION

Gérard GIRAUDON
Rachid DERICHE

Juin 1991

On Corner and Vertex Detection [1]

Sur la détection des jonctions et des points anguleux

Gérard GIRAUDON and Rachid DERICHE

INRIA Sophia Antipolis
2004 Route des Lucioles - 06561 VALBONNE Cedex
FRANCE

April 1991

Programme 4: Robotique, Image et Vision

[1]This work has been partially made under PRC *Orasis* contract and Esprit Project P2502 *Voila*.

Abstract

In a previous paper [Der90], an accurate scale-space based corner detector has been proposed. It has been derived from an analytical study that clarifies completely the behavior of some well known approaches used to detect corners. This paper is an extension of the work to the problem of vertices characterization and detection in real images. We first deal with an analytical study for a general vertex model that allows to better understand the behavior of this important feature in the scale-space. In particular, we show that a trihedral vertex has two elliptic maxima on extremal contrast surfaces if the contrast is sufficient, and this allows to classify vertices in 2 classes: "vertex" and "vertex as corner". The corner detection approach previously developed is applied to accurately detect trihedral vertices. A test is executed to make a distinction between vertex and corner among detected features. Several promising experimental results given in the last section have been carried out using noisy synthetic and real images including corners and vertices.

Résumé

Dans un papier précédent [Der90], nous avons proposé un détecteur de points anguleux se basant sur une analyse multi échelle et permettant une localisation exacte de ces points. Le formalisme adopté nous a permis de faire une étude analytique clarifiant complètement le comportement des détecteurs de points anguleux utilisés jusqu'a présent, et montrant l'impossibilité qu'ils ont à délivrer une localisation exacte du point. Ce rapport est une extension de ce travail au problème de la caractérisation des jonctions et de leur détection dans les images. Nous commençons tout d'abord à faire une étude analytique à partir d'un modèle très général de jonction trihédrale (jonction à trois branches) ce qui nous permet de bien comprendre son comportement dans l'espace multi échelle. En particulier, nous montrons que ce type de jonctions a toujours deux maxima elliptiques positionnés sur les surfaces de contraste extrème et cela uniquement si le contraste est suffisant. De plus, nous montrons que le passage d'un point anguleux à une jonction se fait continument. Cela nous permet alors de classer les jonctions en deux classes : les "jonctions" et les "jonctions-coin". Ensuite, nous montrons que l'on peut utiliser le détecteur de point anguleux [Der90] pour détecter précisément à la fois les points anguleux et les jonctions trihédrales. Un test est effectué a posteriori pour la distinction entre ces deux caractéristiques. Enfin, de très nombreux résultats sur des images de synthèse bruitées ainsi que sur des images réelles sont présentés, résultats montrant la précision de localisation et l'intérêt de l'approche.

1 Introduction

In scene analysis, the goal of low level vision is image segmentation, in particular feature extraction. One of the most popular technique is edge extraction. An expansive literature has been developed from Marr and Hildreth's works [Mar80] to Canny or Deriche's works [Can86], [Der87] based on the first or second derivatives of images. These works use a very simple model for the edge model : the step edge or Heaviside function. This model, essentially one-dimensional, cannot give access directly to another class of important features in scene analysis : corners or vertices. To obtain these features from edges, line segments from polygonal approximation or curvature analysis along edge chains must be used [Asa86], [Der88], [Med86].

Detection of corners or vertices is very important because these features represent a relevant information in computer vision. These features are often used to identify objects in the scene or used for stereoscopic matching or displacement vector measuring etc... So, accurate localization of these features is of great interest.

An alternative method can be used to detect these features directly from a gray-level image. So, several techniques have been proposed in this way. These techniques are based either on heuristic techniques like the "interest operator" of Moravec [Mor77] or on the measurement of the gradients and of the curvatures of the surface. In this second type and for corner detection, we find works of Dreschler and Nagel [Dre82], [Nag83], Kitchen and Rosenfeld [Kit82] and more recently works of Noble [Nob88] Harris and Stephen [Har88] and Guiducci [Gui88].

In a recent paper [Der90], we have presented an analytical study, in the second type of approach, to detect corners. Our approach is based on zerocrossing of Laplacian and a differential measure named DET [Bea78] in multi scale space. The main characteristic of this paper is to give a formal representation of corner detection. In particular, this study has been allowed us to know exactly what is the behavior of some classical measures like the one proposed by Nagel or Kitchen and to correct their faulty localization.

In this paper, we extend our approach to vertex detection and we show that this approach is a general methodology for an accurate detection of discrete surface intersection. As corner detection, we make an analytical study based on vertex model. This study allows to classify vertex in 2 sets in function of the number of DET maxima Finally, we present results of robustness of detector on synthetic noisy image with a very low ratio SNR and on real data.

The paper is organized as follows. The first section starts with the presentation of some previous works on corner and vertex detection. The second section is devoted to the presentation of an analytical study that deals with the vertex detection problem, and allows us to better understand what happens around such features. A last section is then devoted to experimental results on synthetic and real images. An

appendix gives some elementary results on surface differential geometry needed to the comprehension of this report.

2 Previous Works

In this section, we briefly review a few classical feature detectors presented in the literature on corners and vertices. Our review is neither intended to be exhaustive nor does it aim to present corner-vertex detectors in detail (in general, there are many papers on corner detection and a few papers on vertex detection). Several approaches to the problem of detecting feature points have been reported in the literature in the last few years. They can be broadly divided into two groups.

The first group involves first extracting edges as a chain code and then searching for points having maximal curvature as corner point. Maximal curvature points can be found as following :

- At every chain code point, the unit vector tangent is estimated by using pixel coordinates of chain code (see [Asa86], [Med86] or [Mok86])
- A every chain code point, the unit vector tangent is estimated by using the partial derivatives of image $I(x, y)$ with respect to x and y (see [Der88]).

For vertex detection, people use polygonal approximation after chaining and compute the vertex location as the intersection of three segments whose extremities are in a window of size NxN. A interesting work [Bey89] which uses segment grouping can be read for this technique.

The second group consists of approaches that work directly at the gray scale level. Gray scale level based corner detectors are then used to compute locally a measure of *cornerness* C defined as the product of gradient magnitude and the rate of change of gradient direction. Corners are then obtained by thresholding. Among the most popular corner detectors are those proposed by : Beaudet [Bea78], Dreschler and Nagel [Dre82], Kitchen and Rosenfeld [Kit82], and Zuniga and Haralick [Zun83]. In fact, it has been reported by Nagel [Nag83] and by Shah and Jain [Sha84] that the three last detectors are equivalent. We have shown in [Der90] that these detectors cannot detect the exact position of corner. A more recent approach is the one developed by Harris and Stephens [Har88] and Noble [Nob88]. De Michelli *et al* [Mic89] present a comparative study between zero crossing and gradient approaches on corner and trihedral vertex detection. One of the conclusions is that their scheme cannot correctly detect corners or vertices. However, they remark that a vertex V is a zero crossing at every scale and its Gaussian curvature is always negative (hyperbolic type).

Now, we present in details, Beaudet's approach for corner detection. Then we will present our own approach on this problem. For a large comment of the other methods on corner detection, see [Der90].

In 1978, Beaudet [Bea78] proposed a rotationally invariant operator called DET. If $I(x,y)$ is the intensity surface of an image, then:

$$DET = I_{xx}I_{yy} - I_{xy}^2 \tag{1}$$

The corner detection is based on the thresholding of the absolute value of the extrema of this operator. In fact, this operator can be interpreted as the determinant of the second derivatives **Hessian**,

$$\mathbf{H} = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix} \tag{2}$$

which yields the product of the principal curvatures $\kappa_{min}\ \kappa_{max}$, called the Gaussian Curvature [Lip69]. More exactly, for an intensity surface :

$$(\kappa_{min}\kappa_{max}) = \frac{DET}{(1 + I_x^2 + I_y^2)^2} \tag{3}$$

In terms of differential geometry (see Appendix), we can say : for a pixel $I(x,y)$,

If $\kappa_{min}\kappa_{max} > 0 \Longleftrightarrow$ the pixel is an elliptic point

If $\kappa_{min}\kappa_{max} < 0 \Longleftrightarrow$ the pixel is a hyperbolic point

If $\kappa_{min}\kappa_{max} = 0 \Longleftrightarrow$ the pixel is a parabolic point

DET and Gaussian Curvature have the same sign because the denominator of equation (3) is always positive. This means near a corner, DET gives an elliptic and hyperbolic parts (respectively a positive and negative response) on both side of the edge. In a recent paper [Der90], we have stressed an interesting characteristic : the location of an elliptic maximum point is *always inside the corner*, independently of the local image contrast. Moreover near a corner, DET has an elliptic maximum (positive maximum in all directions) and but has not a hyperbolic maximum (negative minimum in all directions). DET has a hyperbolic maximum only in a particular direction, but this elliptic point detection does not allow to locate the corner accurately.

In the same way, for a trihedral vertex composed of three surfaces A,B and C, DET gives elliptic and hyperbolic parts. But in this case, the location of these parts depends on the contrast between A,B and C. This property will be illustrated in section 3 and will be used to classify vertices in two classes.

In our last paper [Der90], we have proposed an analytical study for a accurate corner detection. This detector is based on the two following remarks :

- The exact localization of corner is always given by a zero crossing in scale space i.e. it does not move when the size of the Gaussian filter varies as noted by [Ber84] and [Mic89].

- Near a corner, a elliptic maximum of DET exists and its location moves in scale space. But its trajectory is a line which passes through the location of corner. In this case, this line is exactly the bisector of the corner angle. We have found that the local maximum in all the directions for such a surface is located exactly in the bisector line at ($x = 1.17134\sigma, y = 1.17134\sigma$) for a right angle. Therefore, detecting the corner point as the point where this local maximum occurs leads to a displacement between the true point and the one detected equal to 1.6565σ. Dealing with an angle of $\frac{\pi}{4}$ leads to displace the position of the local maximum to ($x = 2.58231\sigma, y = 1.06963\sigma$) (i.e in the bisector line of the angle). The displacement from the true corner point is in this case equal to 2.795σ.

So, we have proposed the following detector :

- Using equation
$$DET = I_{xx}I_{yy} - I_{xy}^2 \tag{4}$$
compute two images DET_1 and DET_2 corresponding to two different values $\sigma_1 < \sigma_2$ for the scale space.

- Detect and threshold all the local maxima (only positive value) of DET_1 and DET_2.

- For each position (x_2, y_2) of a local maximum in DET_2, search for a local maximum in DET_1. The search is performed along a spiral centered on (x_2, y_2), limited to a 7x7 window and we take the first local maximum (x_1, y_1) which is found. In order to improve the precision of the localization, we fit a quadric around each local maximum (x_i, y_i) and select the corrected position of each local maximum with sub-pixel precision.

- Compute the line equation joining (x_1, y_1) and (x_2, y_2). This gives the estimated bissector line where the exact corner position must lie. Along this line, starting from (x_2, y_2) and moving in an opposite direction to (x_1, y_1), validate the first zero-crossing location (X, Y) of the Laplacian as the exact localization of the corner at pixel resolution (if we get the minimum absolute value of Laplacian pixel), or in sub-pixel resolution (if we track the exact zero-crossing by a curve fit on laplacian).

In the next section, we apply the corner detection methodology for vertex detection. From a vertex modelisation, we observe the behavior of equation (4) in scale space.

3 A General trihedral Vertex Model and its Analytical study

In this section, we consider a general trihedral vertex model and study its behavior in the scale space. This allows us to derive results that clarify completely the behaviour of the surface described by the intensity function of a Gaussian filtered trihedral vertex. The elements of the first and second fundamental form of the intensity surface function of the trihedral vertex are calculated analytically and some useful measure as the determinant of the Hessian are then derived. In particular, we will show that depending of the contrast between the different parts of the trihedral vertex, there are one, two or three local extrema associated to the determinant of the Hessian. These local maxima move in the scale-space along a line that passes through the exact position of the central point of the vertex (i.e 0,0). Based on this study, we will present an approach that allows us to characterize and extract with good accuracy the central point of the vertex. This approach makes use of two important properties of the vertex in the scale space. First, the property that the Laplacian image is zero at the exact position of the vertex and second a property associated to the measure we propose to use in order to get the vertex. It will be shown that vertex extracted using this measure moves in the scale space on a line passing through the exact position of the vertex. We then combine these properties in order to get the exact position of the vertex.

3.1 Notations and definitions

We introduce here some functions that will be largely used in the rest of the paper.

Let $g(x)$ denote the zero-mean Gaussian filter :

$$g(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{5}$$

The two-dimensional Gaussian filter G can be expressed as :

$$G(x, y) = g(x)g(y) \tag{6}$$

Following Berzins [Ber84], we work in coordinate system where the unit length is equal to the scale factor σ of the filter. In order to convert the results into a more general coordinate system (X, Y), we use the following transformation:

$$\begin{cases} x = \frac{X}{\sigma} \\ \\ y = \frac{Y}{\sigma} \end{cases} \tag{7}$$

Let Φ denote the error function given by :

$$\Phi(x) = \int_{t=-\infty}^{x} g(t)dt \tag{8}$$

Let U define the unit step function

$$U(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The output of the 2D Gaussian filter G for a 2D input function $I(x,y)$ can be computed by evaluating the following convolution integral

$$S(x,y) = \int_{\alpha=-\infty}^{+\infty} \int_{\beta=-\infty}^{+\infty} G(\alpha,\beta)I(x-\alpha, y-\beta)d\alpha d\beta \tag{10}$$

3.2 A General Trihedral Vertex Model

A general trihedral vertex as shown in Figure 3.1 can be modeled by the following 2D intensity function.

$$I(x,y) = \begin{cases} A & \text{if } x \leq 0 \quad \text{and } y < m'x \\ B & \text{if } x > 0 \quad \text{and } y < mx \\ C & \text{if } y \geq mx \quad \text{and } y \geq m'x \end{cases} \tag{11}$$

where $m = tan(\theta)$ and $m' = -tan(\phi)$.

From this model, we can derivate the corner model given in [Der90] with A=B and $\theta = 0$.

Using the step function $U(x)$ defined by (9) yields the following expression for $I(x,y)$:

$$I_{(\theta,\phi)}(x,y) = AU(-x)U(m'x-y) + BU(x)U(mx-y) + CU(y-mx)U(y-m'x) \tag{12}$$

If we convolve this 2D intensity function with the 2D Gaussian filter given by (6) we get the following filtered image $S(x,y)$:

$$\begin{aligned} S(x,y) = A + C(\textstyle\int_{-\infty}^{x} g(\alpha)\Phi(y-m(x-\alpha))d\alpha + \int_{x}^{\infty} g(\alpha)\Phi(y-m'(x-\alpha))d\alpha) \\ -A(\textstyle\int_{x}^{\infty} g(\alpha)\Phi(y-m'(x-\alpha))d\alpha) - B(\int_{-\infty}^{x} g(\alpha)\Phi(y-m(x-\alpha))d\alpha) + (B-A)\Phi(x) \end{aligned} \tag{13}$$

This equation describes a Gaussian filtered version of the general trihedral vertex given by (12) with angle θ and ϕ and intensity amplitude of A,B and C on each part. Figure 3.2 illustrates such surface for the case where $\theta = \pi/3$, $\phi = \pi/4$, $A = 1$,$B = 2$ and $C = 3$.

In order to deal with an analytical expression for the different elements of the first fundamental form of the intensity surface described by the trihedral vertex, we have to get the components $S_x(x,y)$ and $S_y(x,y)$ of the gradient vector $\vec{\nabla}S(x,y)$.

Using the following changes of variables:

$$\begin{aligned} u &= xsin(\theta) - ycos(\theta) \\ v &= xcos(\theta) + ysin(\theta) \\ \bar{u} &= -xsin(\phi) - ycos(\phi) \\ \bar{v} &= xcos(\phi) - ysin(\phi) \end{aligned} \tag{14}$$

The components $S_x(x,y)$ and $S_y(x,y)$ of the gradient vector $\vec{\nabla}S(x,y)$ can be calculated after some manipulations and put in the following analytical and simplified form :

$$\vec{\nabla}S(x,y) = \begin{bmatrix} (B-A)g(x)\Phi(-y) - (A-C)g(\bar{u})\Phi(-\bar{v})sin(\phi) + (B-C)g(u)\Phi(v)sin(\theta) \\ (C-B)g(u)\Phi(v)cos(\theta) + (C-A)g(\bar{u})\Phi(-\bar{v})cos(\phi) \end{bmatrix} \tag{15}$$

An analytical expression for the norm of the gradient vector can then easily be calculated. The 3D plot of $\| \vec{\nabla}S(x,y) \|$ leads to see that gradient magnitude decreases near the origin point and affects the edge extraction process as it will be shown in the next subsection.

As seen in the previous section devoted to the differential geometry, the Hessian matrix is important for the description of surfaces. The elements of such a matrix can be calculated by deriving two times in the x and y direction the surface $S(x,y)$. Applying this to our vertex model surface yields after some manipulations the following elements:

$$\begin{aligned} &S_{xx}(x,y) = (A-B)xg(x)\Phi(-y) + (C-A)g(\bar{u})\sin(\phi)(\Phi(-\bar{v})\bar{u}\sin(\phi) - g(-\bar{v})\cos(\phi)) \\ &+(B-C)g(u)\sin(\theta)(g(v)\cos(\theta) - u\Phi(v)\sin(\theta)) \\ &S_{xy}(x,y) = S_{yx}(x,y) = (A-B)g(x)g(y) + (C-A)\sin(\phi)(\bar{u}g(\bar{u})\Phi(-\bar{v})\cos(\phi) \\ &+g(x)g(y)\sin(\phi)) + (B-C)\sin(\theta)(ug(u)\Phi(v)\cos(\theta) + g(x)g(y)\sin(\theta)) \\ &S_{yy}(x,y) = (C-B)\cos(\theta)(ug(u)\Phi(v)\cos(\theta) + g(x)g(y)\sin(\theta)) + (C-A)\cos(\phi) \\ &(\bar{u}g(\bar{u})\Phi(-\bar{v})\cos(\phi) + g(x)g(y)\sin(\phi)) \end{aligned} \tag{16}$$

It is worth noticing that letting $\theta = \phi$ and $C = 0$ leads to the particular results obtained for the symmetric trihedral vertex studied by Michelli *et al* [Mic89] in their interesting contribution to the study of the localization in edge detection. In fact, this part of our study generalizes a part of the work addressed by Michelli [Mic89].

3.3 On the extraction of the edges of a trihedral vertex

It is well known that edges can be extracted from an image using the non-maximum suppression scheme [Can86], [Der87] or the zero-crossing scheme [Ber84]. In this subsection, we briefly remind the main difference between both approaches in the case where general vertices have to be detected (see [Mic89] for the particular case of a symmetric vertex). In the first approach, edges are extracted as local maxima of the gradient magnitude in the gradient direction. This is equivalent to extract points where the second directional derivative of the gradient magnitude image along the gradient direction is equal to zero. An explicit representation of the second directional derivative in the gradient direction $\mathbf{n}$ can be found to be :

$$\frac{\partial^2 S}{\partial \mathbf{n}^2} = \frac{S_{xx}.S_x^2 + 2.S_x.S_y.S_{xy} + S_{yy}.S_y^2}{(S_x^2 + S_y^2)} \tag{17}$$

The location of the zero-crossings of (17) corresponds to the location of the discontinuity in the 2D step filtered function (13) extracted following the non-maximum suppression scheme. Figure 3.3 displays the curve that represents the set of points extracted through the use of this measure in the case of a vertex with $\theta = \pi/3, \phi = \pi/4, A = 1, B = 2$ and $C = 3$. This figure illustrates a rounding effect of the edge extraction with the non-maxima suppression scheme. It is worth noticing that we do not have a zero crossing for the central point $\frac{\partial^2 S}{\partial \mathbf{n}^2}(0,0) \neq 0$.

In the second approach, edges are extracted as zero-crossings in the Laplacian image given by :

$$\nabla^2 S(x,y) = S_{xx}(x,y) + S_{yy}(x,y) \tag{18}$$

Dealing with the vertex described by (13) leads to the following expression for the Laplacian :

$$\begin{aligned}
&\nabla^2 S(x,y) = (A-B)xg(x)\Phi(-y) \\
&+(C-A)g(\bar{u})\sin(\phi)(\Phi(-\bar{v})\bar{u}\sin(\phi) - g(-\bar{v})\cos(\phi)) \\
&+(B-C)g(u)\sin(\theta)(g(v)\cos(\theta) - u\Phi(v)\sin(\theta)) \\
&+(C-B)\cos(\theta)(ug(u)\Phi(v)\cos(\theta) + g(x)g(y)\sin(\theta)) \\
&+(C-A)\cos(\phi)(\bar{u}g(\bar{u})\Phi(-\bar{v})\cos(\phi) + g(x)g(y)\sin(\phi))
\end{aligned} \tag{19}$$

Equation (19) can be solved numerically to find the contours where the Laplacian is equal to zero. The resulting contours are shown in Figure 3.4 for a Gaussian filtered vertex with the following set of parameters :

- Figure 3.4.a : $\theta = \pi/3, \phi = \pi/4$ and A=1,B=2,C=3.
- Figure 3.4.b : $\theta = 0, \phi = 0$ and A=2,B=1,C=0
- Figure 3.4.c : $\theta = 0, \phi = 0$ and A=1,B=-1,C=0

Let us remember that both x and y are given in units of standard deviations σ of the Gaussian filter. This illustrates very well the rounding effect due to the use of the Laplacian instead of the non-maxima suppression scheme.

From these remarks, an important point to notice is that the Laplacian allows to recover exactly the position of the vertex since it can easily be checked that $\nabla^2 S(0,0) = 0$ (i.e the spatial position of a general vertex does not change in the scale space). This is not the case for the non-maximum suppression scheme. We will make use of this important point in order to correct the position of the extracted vertex.

Note : For a particular corner case with C=B and $\theta = \phi = 0$, equation (19) becomes

$$\nabla^2 S(x,y) = (A-B)(xg(x)\Phi(-y) + yg(y)\Phi(-x)) \tag{20}$$

3.4 On DET extrema extraction of a trihedral vertex

Now, the behavior of the DET equation (1) is analyzed. In our formalism, we can write this equation as following :

$$DET(x,y) = S_{xx}(x,y)S_{yy}(x,y) - S_{xy}(x,y)^2 \tag{21}$$

Replacing the different second derivatives that appear in the expression of the DET by their expression given (16), we get a complex expression for the DET that depends on :

- contrast values : value(A-B), (C-A) and (B-C)
- angle values : ϕ and θ.

For the particular case where $\phi = \theta = 0$, the equation becomes :

$$\begin{aligned} DET(x,y) &= (A-B)g(x)g(y)(xy\Phi(-y)((B-C)\Phi(x) \\ &+(A-C)\Phi(-x)) - (A-B)g(x)g(y)) \end{aligned} \tag{22}$$

For the particular corner case where $\phi = \theta = 0$ and B=C, we obtain a corner [Der90] and the equation becomes :

$$DET(x,y) = (A-B)^2 g(x)g(y)(xy\Phi(-x)\Phi(-y) - g(x)g(y)) \quad (23)$$

In this case, DET(0,0) is always negative (i.e. it is a hyperbolic point) and its value moves in scale space (i.e. depends on σ).

The DET behavior will strongly depend on contrast between A,B and C. To illustrate this fact, we show on figures 3.5, 3.6 and 3.7 plots of DET versus contrast (with $\phi = \theta = 0$)

- For $A > C > B$, Figure 3.5.a shows a 3D plot of DET and figure 3.5.b shows the level curves of surface ($A = 1, B = -1, C = 0$).
- Figure 3.6 shows the level curves of surface for $A > B > C$ ($A = 2, B = 1, C = 0$).
- Figure 3.7 shows the level curves of surface for $A >> B > C$ ($A = 10, B = 1, C = 0$)

Some comments can be made on these results :

- Vertex point ($x = y = 0$) is always an hyperbolic point.
- In function of contrast, DET presents one or two elliptic parts and only one hyperbolic part. For the elliptic part, we have always one or two elliptic extrema (maxima in all direction). The location of these extrema is always *inside* the extremal contrast surface. For example, in 3.5 the elliptic maxima are in A and B and in figure 3.6 they are in A and C. In the case 3.5, the maxima are found in $x = -1.32\sigma, y = -1.195\sigma$ (location in surface A) and $x = 1.32\sigma, y = -1.195\sigma$ (location in surface B) The case presented in figure 3.7 is very interesting. It means that if we have only a big contrast (A >> B and B ≡ C), the behavior of vertex looks like the behavior of corner ([Der90]). So only one elliptic maximum exists and it is located on surface A.
- Hyperbolic parts are less stable. In the case of figure 3.5, we have an extremal hyperbolic point located in vertex (0,0) but the hyperbolic part is on surface A,B and C with the most proportion in surface C (intermediate contrast). In the case of figure 3.6, two hyperbolic extrema exist on B and C surface and in the case 3.7 there is not hyperbolic maximum.
- These results have been verified with other values of ϕ and θ.

As corner detection, we have studied the behavior of elliptic maxima in scale space. And we have plotted the curve which passes through the position of this maximum and the position of vertex. So,

- Fig 3.8. presents this curve for contrast 1 ($A = 1, B = -1, C = 0$).
- Fig 3.9 presents this curve for contrast 2 ($A = 2, B = 1, C = 0$).
- Fig 3.10 presents this curve for contrast 3 ($A = 10, B = 1, C = 0$).

These figures show that elliptic maxima localization move in scale space for the three types of contrast along a line which passes through the vertex. This means that the approach developed for the corner detection can also be applied for the vertex detection.

Moreover, some comments can be made on these curves. First, the curves presented above are not always the bisector of angles (ϕ and θ). Second, we can class vertices in two classes function of contrast and so function of the number of elliptic maxima existing around the vertex. If two maxima exist (case 1 or 2), we have what we call "real or robust vertex" and if only one maximum exists (case 3), we have a "vertex like corner".

3.5 Combining informations to extract accurate vertex

Following the analysis that we have done, two important points have to be noted :

- The exact position of a vertex can be detected as a stable zero-crossing in the scale-space.
- The local maximum in Beaudet measure moves in the scale space along a line that passes through the exact position of the vertex point.

So the method used to detect corners presented in [Der90] can also be used to detect vertices backing to the following strategy :

1. First a Laplacian image is calculated.
2. Second, Beaudet's measure at two or more different scales are calculated and a maxima detection (on elliptic parts) in all directions is performed. Around each detected maximum in the image corresponding to the first scale, we look in the second image for the position of the local maximum. Once this second maximum detected, we then look for the exact position of the corner as the point that belongs to the line segments joining the two positions and where

a zero-crossing occurs in the Laplacian image. With this approach, we have combined the two important points that we have mentioned. Experimentally, this approach has been found to be reasonably stable.

So, with this algorithm, we detect corners and vertices. If a distinction must be made between a corner and a vertex, we can use a remark given above. In the case of "real or robust vertex", we have two detections on vertex location because each detection is created by a curve joined two elliptic maxima. So a distinction between vertex and corner can be made easily. But for the second type of vertex ("vertex like corner"), we have just one elliptic maximum curve, and so we will have only on detection on vertex position like a corner detection. So our detector cannot solve this ambiguity and an other local contrast study would be needed. distinction.

An other possible classification is to say :

- A detected feature is named a "vertex" if its localization is created by the existence of two elliptic maxima curves
- A detected feature is a "corner" i its localization is created by the existence of only one elliptic maximum curve.

4 Experimental results

In this section, we give some experimental results obtained on running our corner and vertex detector on noisy synthetic and real images. All low-level processing required in the approach (smoothing and deriving steps) is performed using the Gaussian filter operator and its derivatives. The detection is performed at pixel precision. To find vertex position (X,Y), we validate the first zero-crossing location (X,Y) of the Laplacian at pixel resolution (we get the minimum absolute value of Laplacian pixel). To reach sub-pixel precision, we are in the process of implementing a tracking approach for the exact zero-crossing by a curve fit on laplacian.

4.1 Results on noisy images

An additive Gaussian noise with standard deviation σ has been added to each synthetic image in order to deal with noisy data. The Signal to Noise Ratio (SNR) is computed as follow :

$$SNR_{dB} = 10\log(\frac{\sum\sum f(x,y)^2}{\sum\sum e(x,y)^2}) \quad (24)$$

where f(x,y) denotes the original image and e(x,y) is the noisy one.

In figures 4.1 to 4.4, we present results on trihedral vertex detection. We have synthethized two types of vertices depending on contrast. We have chosen a vertex with an equal angle for θ and ϕ of 0 for the first type and we have taken $A = 150$, $B = 100$ and $C = 0$.

Figures 4.1.a and 4.1.b present DET maxima for $\sigma = 1$ and $\sigma = 2$. We show here that two maxima exist, one on surface A (=150) and one on surface C (=0). The value of the maximum on C is very low with regard to the one on A (a ratio of 5) and then it could be rejected by a threshold. These maxima are aligned. Figure 4.1.c shows the two detections which are localized on vertex position.
For the second type, we have taken $A = 150$, $B = 0$ and $C = 100$. Figures 4.2.a and 4.2.b present DET maxima for $\sigma = 2$ and $\sigma = 3$. We show that again two maxima exist also one on A (=150) and one on B (=0). In this case, the value of the maximum on C is greater than the one of surface A (a ratio of 3). These maxima are not aligned. Figure 4.2.c shows the two detections which are localized on vertex position. (we have the same result with $\sigma = 1$ and $\sigma = 2$.

For the behavior on noisy images, we have considered the first type of vertex. Figures 4.3 and 4.4 present vertex detection with $SNR_1 = 13dB$ and $SNR_2 = 7dB$. In the first case, a Gaussian filtering with $\sigma = 1$ and $\sigma = 2$ allows to detect exactly the position of vertex. (We have made a threshold to keep just one maximum of DET). But with 7dB and taking $\sigma = 2$ and $\sigma = 3$, we have not a good localization and more we have a false detection. Finally, in figure 4.4, we show the best result obtained with $\sigma = 3$ and $\sigma = 7$ with a $SNR = 7dB$.

In summary, we have verified experimentaly the theoretical results. With a correct contrast between surfaces, we have shown in the vertex case that two elliptic maxima of DET exist always on the surfaces of extremal contrast independently of their locations in scene. Their locations depends on the contrast. Robustness of our detector has been demonstrated on a very low SNR with adapted σ in scale space.

Finally, we show in figure 4.5 and 4.6 results on checkboard image with gray level surface equal to 150 and to 0. This does not correspond to our vertex model (the feature is the intersection of four surfaces) but it is an interesting case. In this special case, four DET maxima exist (compute with $\sigma = 1$ and 2, figure 4.5) and the detection of vertices is good (figure 4.6).

4.2 Results on real data

To illustrate the efficiency of our detector, we show on figure 4.7 the location of features detected by our algorithm. In this image (512.512 pixels with 256 grey level), we have corners and vertices. The parameters of this detection are :

- Laplacian computed by a Gaussian filtering with $\sigma = 1$
- DET computed by a Gaussian filtering with $\sigma = 1$ and $\sigma = 2$
- Threshold on DET maxima = 200
- a 7x7 spiral window from σ_2 to σ_1
- a quadratic fitting for subpixel precision on location of elliptic maximum.

In order to illustrate the detector behavior on vertices, figures 4.8 (a, b, c) shows some local zoom of vertices detection. The corrected position derived by our algorithm seems to be quite promising. As we have indicated in the previous section, we detect as vertex features which are created by the intersection of zero crossing and two lines (existence of two elliptic maxima). So, from point of view scene analysis, all vertices which are present in this image are not extracted because some of them are classified as corner (see figure 4.12 for every feature detection)

Remark : Note that decreasing the threshold for the magnitude of maxima would have caused on more features points to be detected. The results obtained by running the algorithm on a triplet of trinocular stereo images are very stable and suggest that the matching process will be much more easier and reliable using these features. We have implemented the algorithm with a complete multiscale representation using $\sigma = 1$ to $\sigma = 6$. We have used a polygonal approximation of the maxima line instead of simply joining two maxima. This implementation does not increase the precision of localization and yields inferior results on noisy images than presented here. This is due to the bad localization with $\sigma = 1$ or 2 and its influence on the mean square process to compute the line.

5 Conclusion

A formal representation of corner and vertex detection has been proposed in this paper. In particular, an analytical study that allows us to know exactly what is the behavior of our detector proposed in [Der90] around trihedral vertices has been developed. We have shown that near three surfaces, two elliptic maxima of DET exist and their location is inside extremal contrast surface. The intermediate surface shows always an hyperbolic minima. We have shown that our detector allows us to find

the exact position of vertex. The approach proposed has been tested on many noisy synthetic data and real images and its robustness seems very promising.

6 Acknowledgments

The authors would like to express their acknowledgments to **Maple** and **Mathematica** for all the mathematical and graphical facilities.

7 Figures and Images

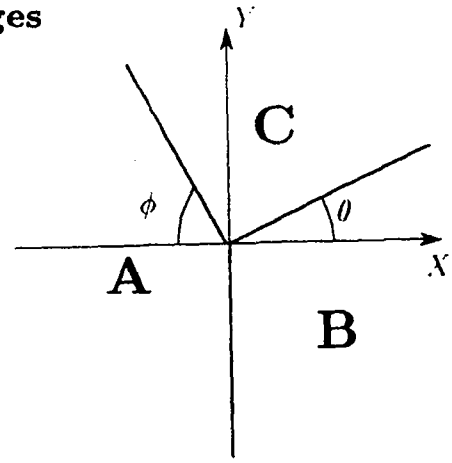


Fig 3.1: A General Trihedral Vertex

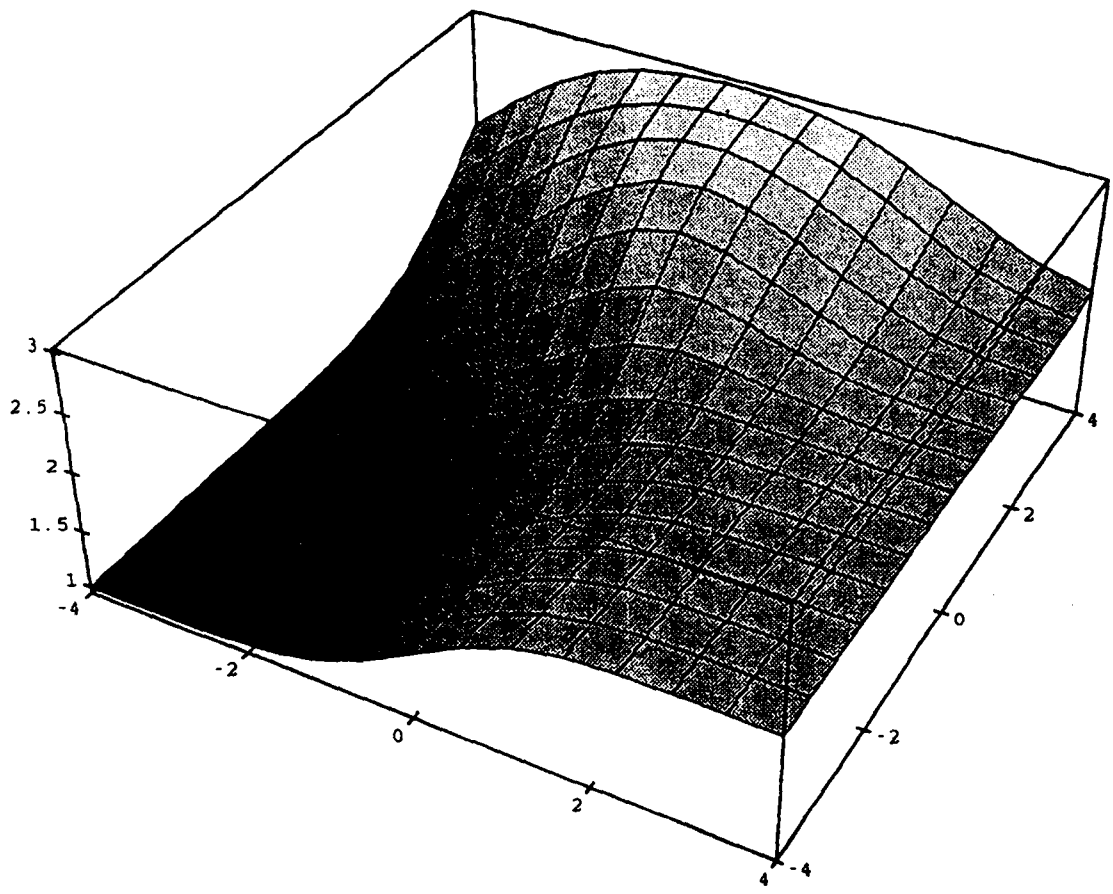


Fig 3.2: Gaussian Filtered Trihedral Vertex

A=1,B=2,C=3,theta=Pi/3,phi=Pi/4

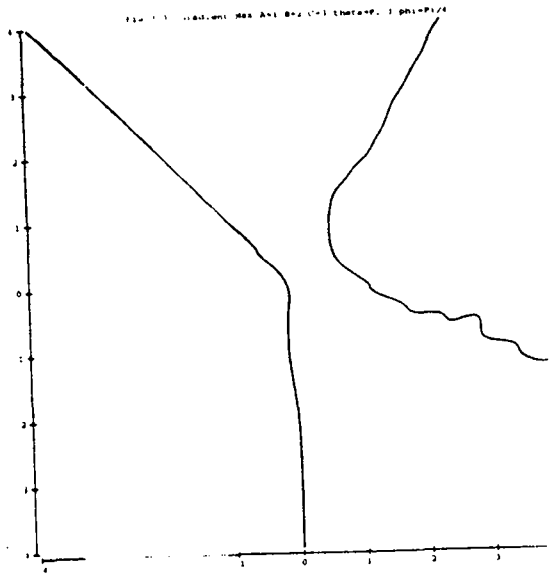


Figure 3.4.b
zero corssing with
$\Theta=\phi=0$: A=2, B=1, C=0

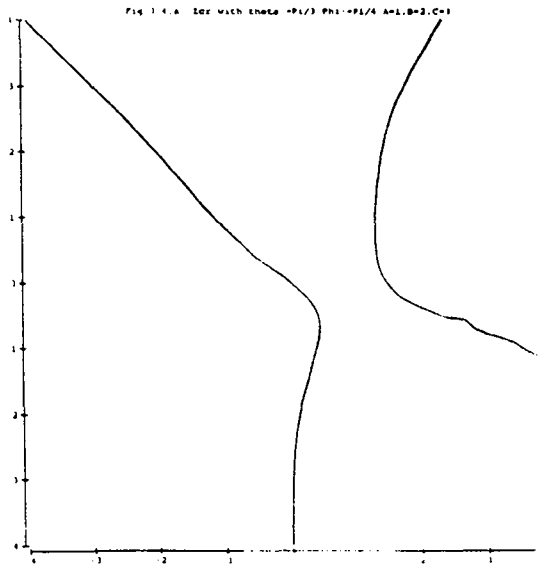
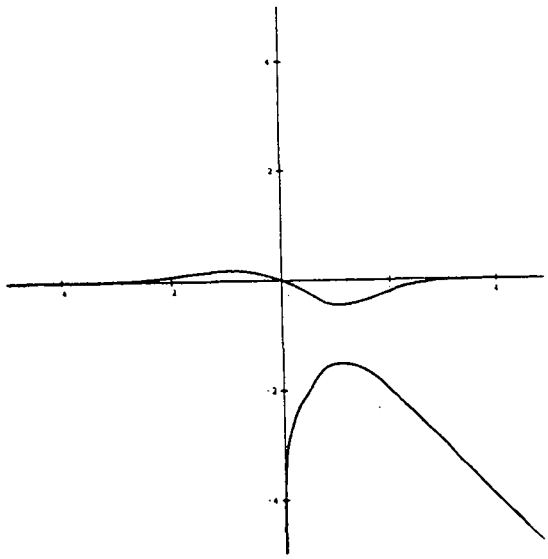
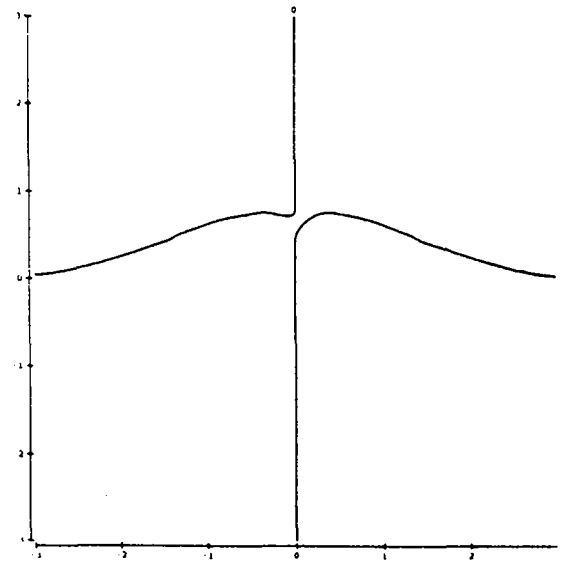


Figure 3.4.c
zero crossing with
$\Theta=\phi=0$: A=1, B=1, C=0





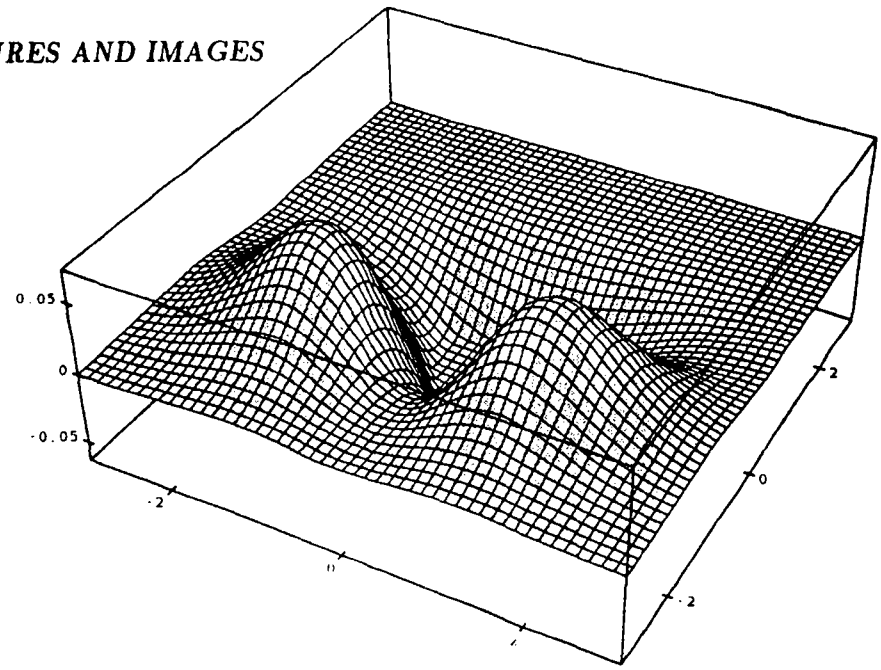


Figure 3.5.a : 3D of DET
(A=1, B=-1, C=0)

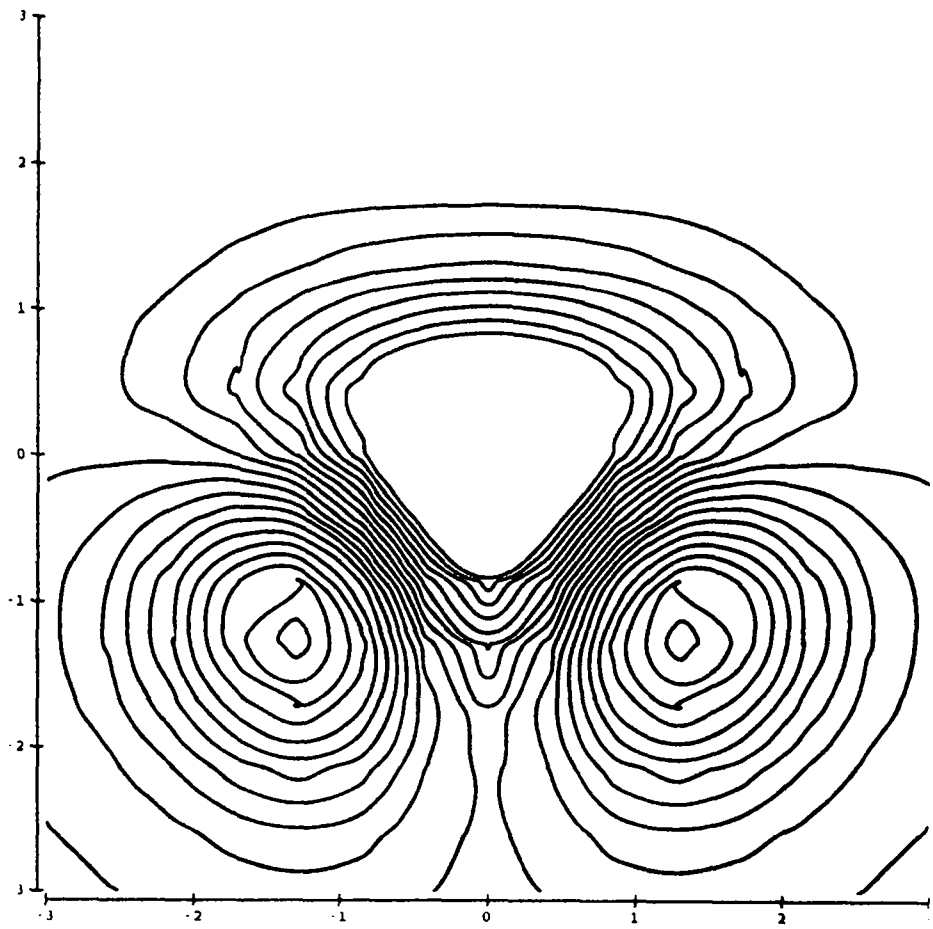
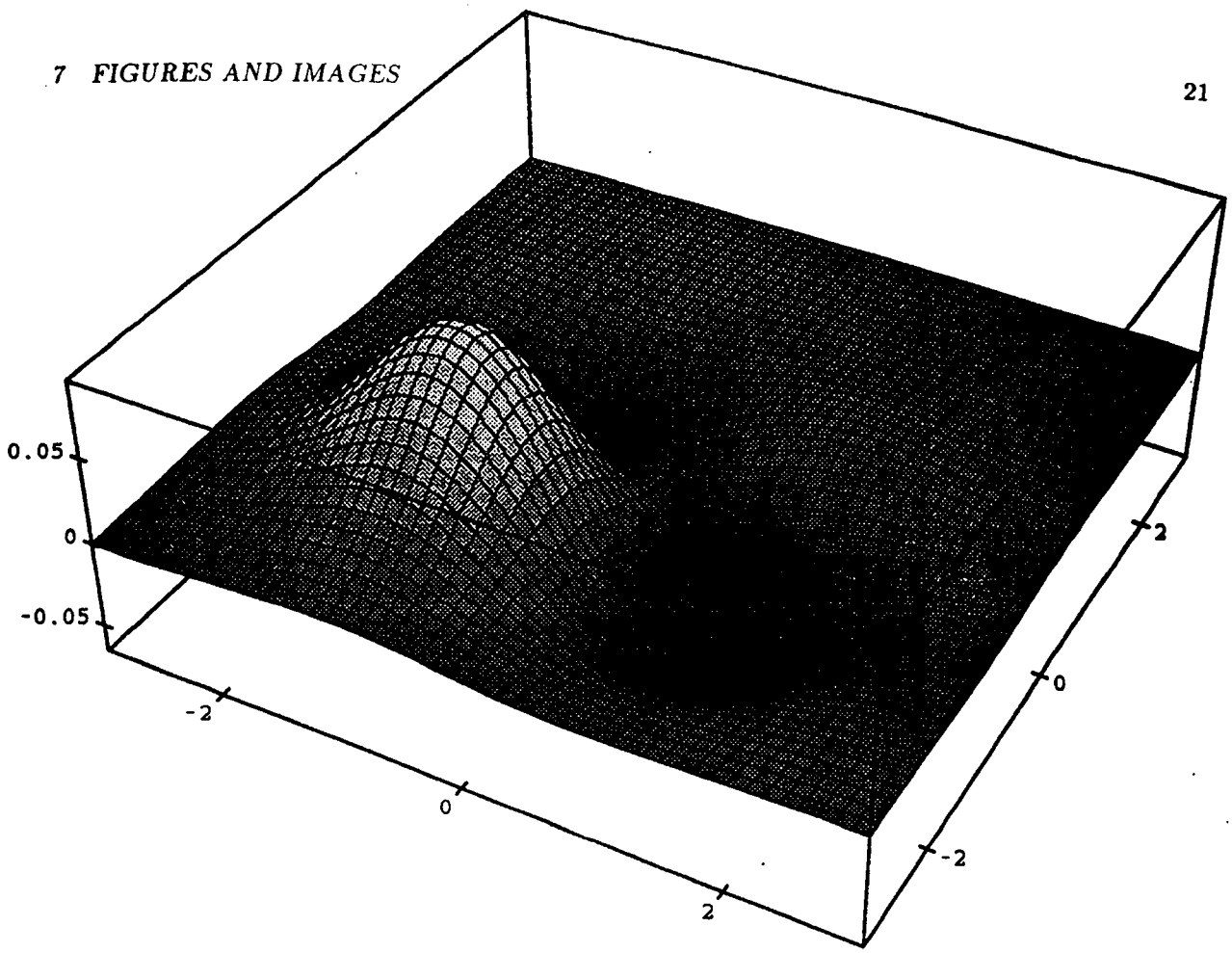


Figure 3.5.b: Level curves of
DET (A=1, B=-1, C=0)



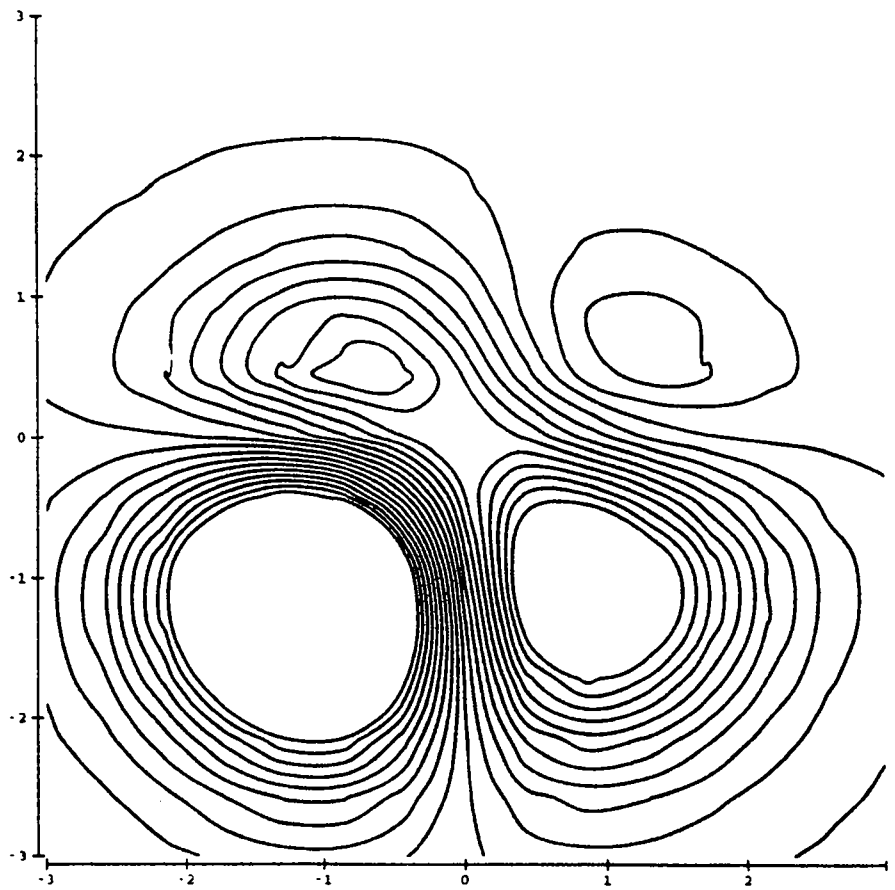
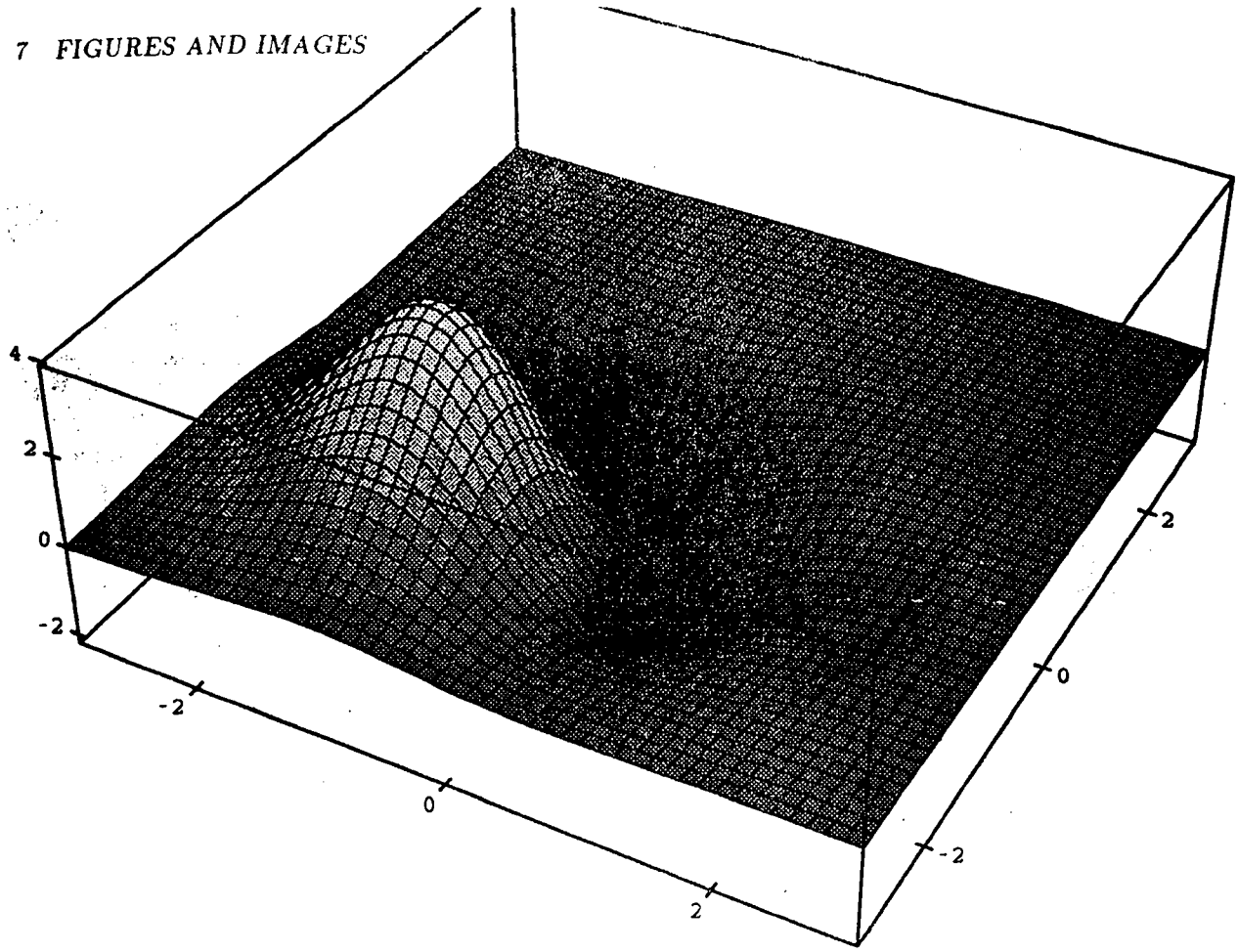


Figure 3.6 : Level curves with
A=2, B=1, C=0



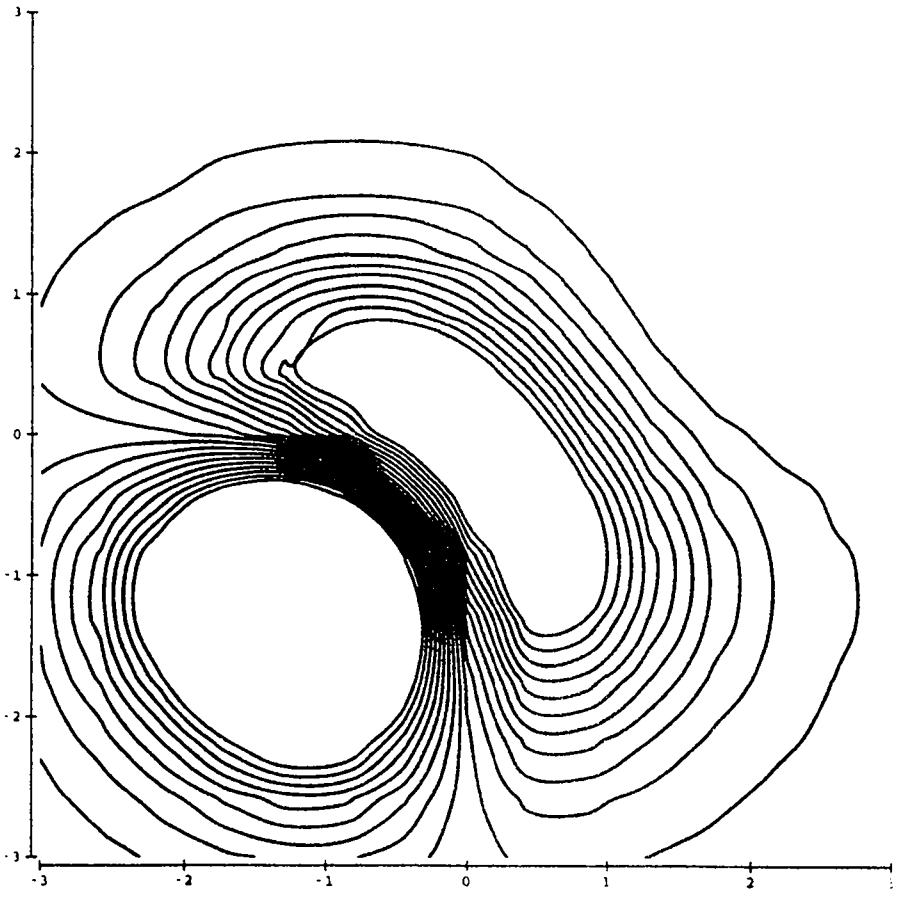


Figure 3.7 : Level curves with
A=10, B=1, C=0

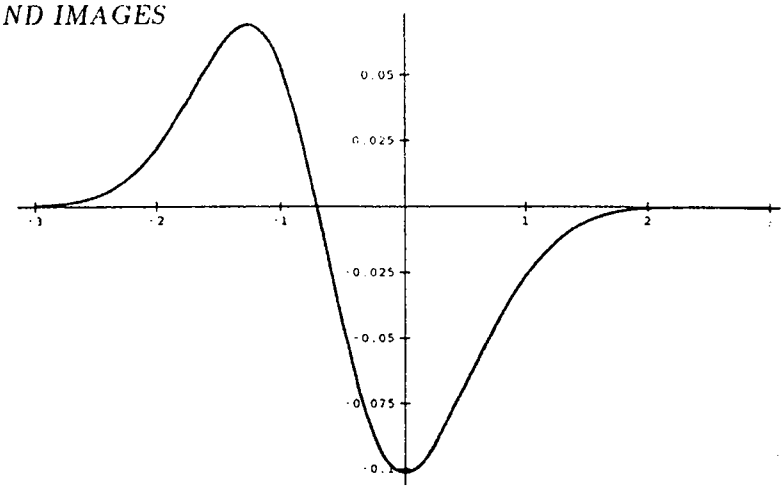


Figure 3.8 : A=1, B=1, C=0

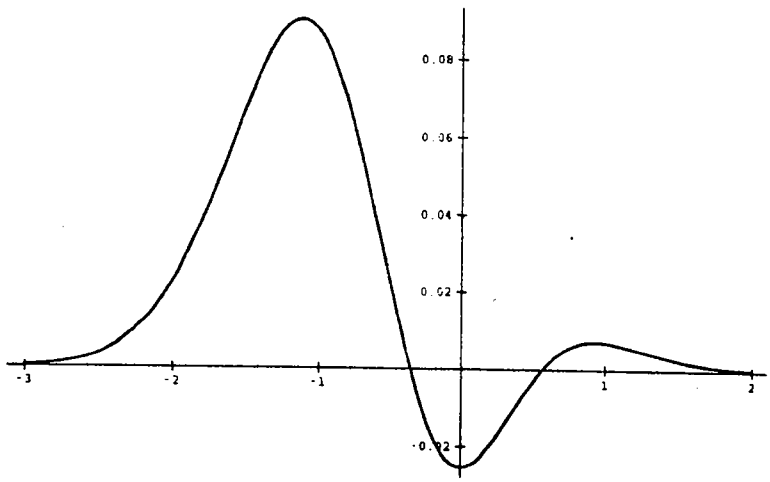


Figure 3.9 : A=2, B=1, C=0

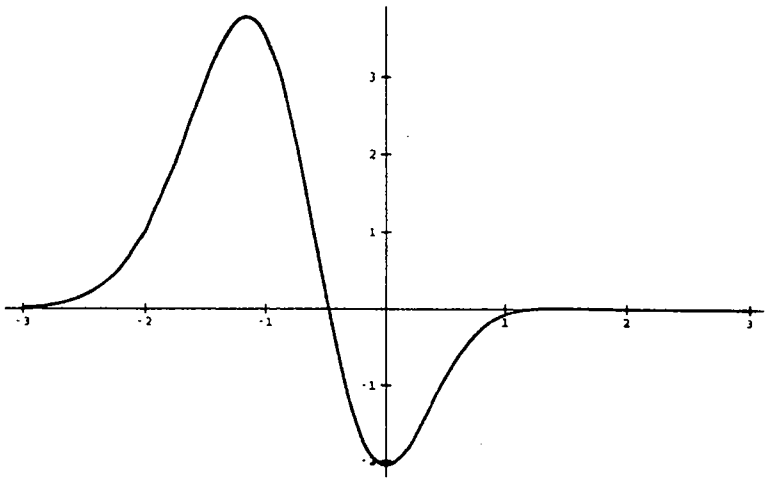
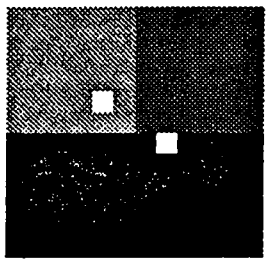
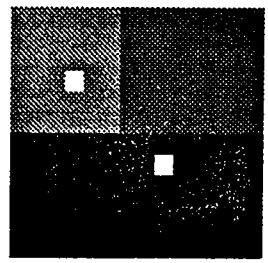


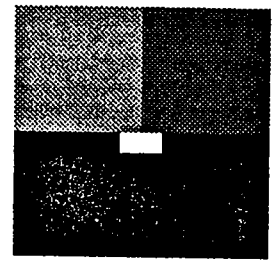
Figure 3.10 : A=10, B=1, C=0



4.1.a : $\sigma = 1$

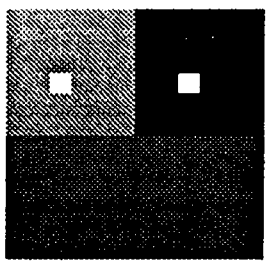


4.1.b : $\sigma = 2$

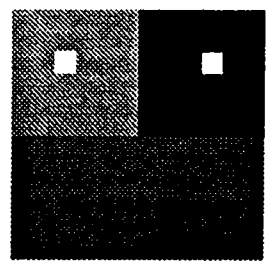


4.1.c : Vertex localization

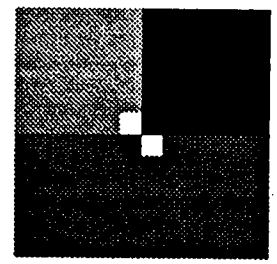
No 4.1 : sub-image with a vertex of type 1



4.2.a : $\sigma = 2$



4.2.b : $\sigma = 3$



4.2.c : Vertex localization

No 4.2 : sub-image with a vertex of type 2

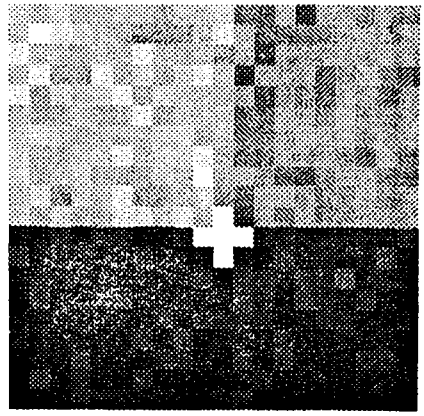


Fig 4.3 : Vertex localization with 13dB, $\sigma = (2,1)$

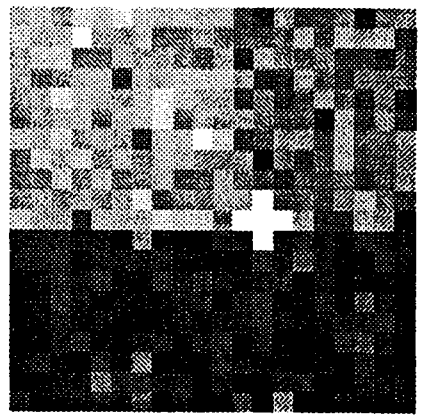


Fig 4.4 : Vertex localization with 7dB, $\sigma = (7,3)$

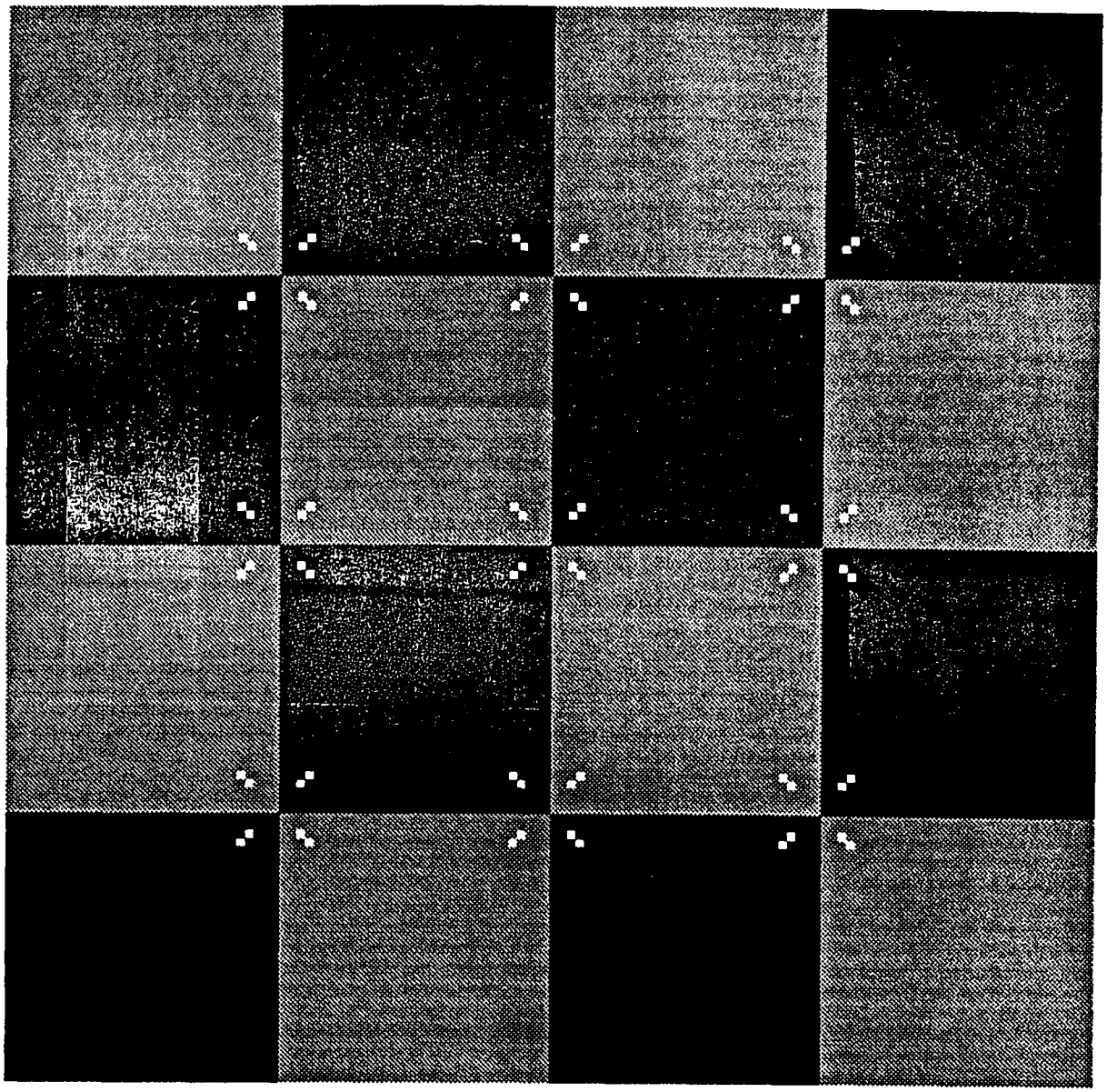


Figure No 4.5 : Ellipic maxima detected with $\sigma = 3$ and $\sigma = 2$

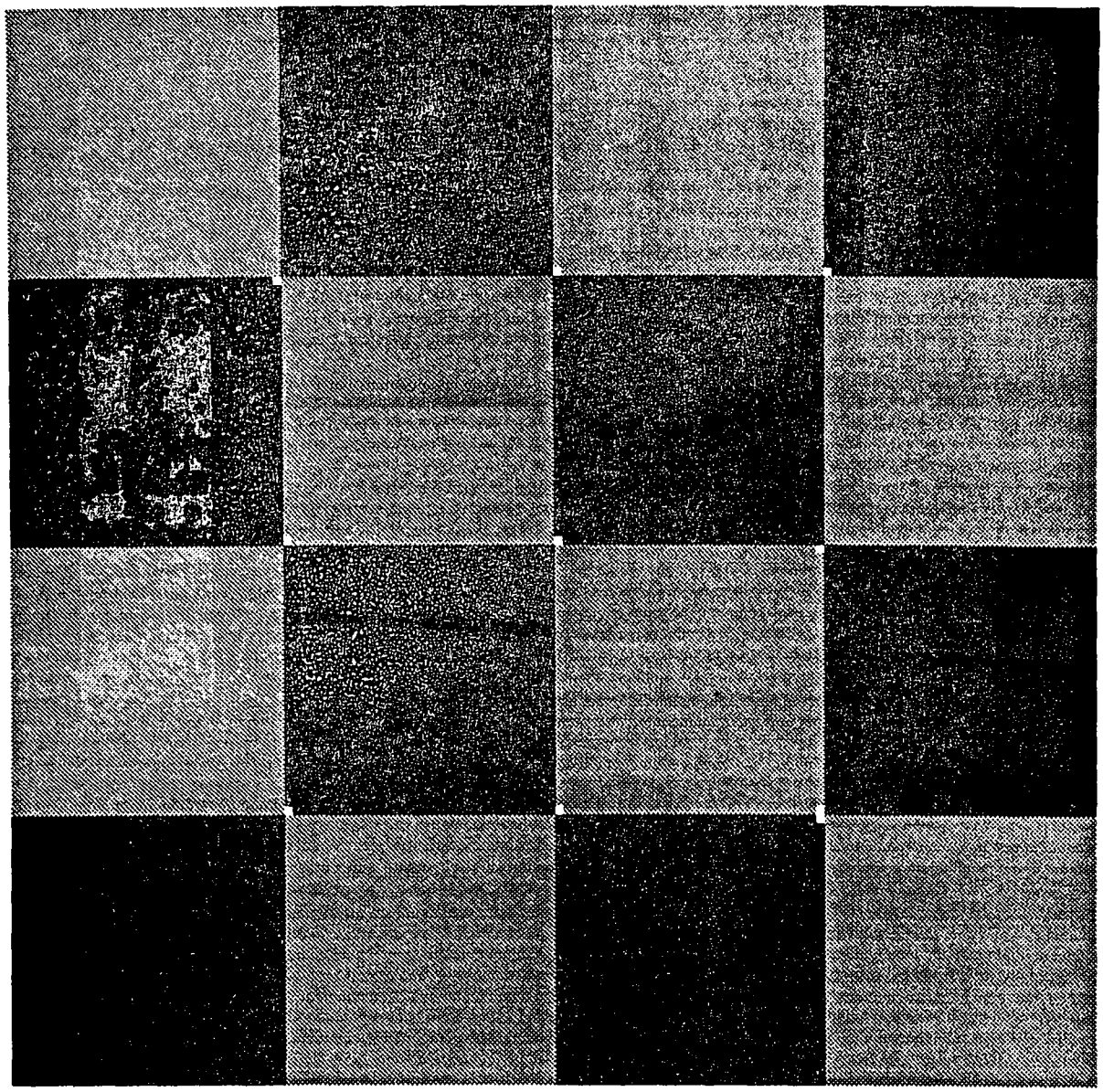
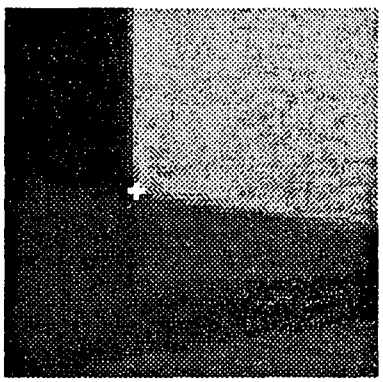
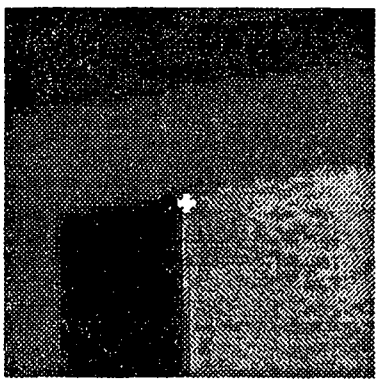


Figure No 4.6 : Vertices detected with $\sigma = 3$ and $\sigma = 2$

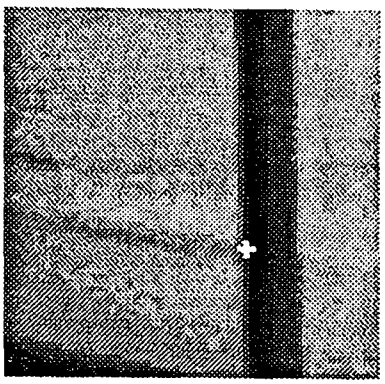
No 4.7 : Features extracted on real image with $\sigma = (2, 1)$



4.8.a : Example 1



4.8.b : Example 2



4.8.c : Example 3

No 4.8 : Vertex localization on real image with $\sigma = (2, 1)$

References

[Asa86] H. ASADA and M. BRADY: "The Curvature Primal Sketch" IEEE PAMI, Vol 8 1986, pp 2-14.

[Bar80] S.T. BARNARD and W.B. THOMPSON : "Disparity analysis of images", IEEE PAMI, Vol2, No 4, July 1980, pp 333-340

[Bea78] P.R. BEAUDET: "Rotational Invariant Image Opereators", in Int, Conf Pattern Recognition 1978 pp 579-583.

[Ber84] V. BERZINS "Accuracy of Laplacian Edge Detectors", in CVGIP Vol27, 1984 pp 195-210

[Bey89] D. J. BEYMER "Junctions : Their detection and use for grouping in images" Master of Science, MIT, May 1989. 177 pages.

[Can86] J.F. CANNY : "A Computational Approch to Edge Detection", IEEE PAMI Vol8, No 6 November 1986 pp 679-698

[Der87] R. DERICHE : "Optimal Edge Detection Using Recursive Filtering" in International Journal of Computer Vision, pp 167-187, 1987.

[Der88] R. DERICHE and O.D. FAUGERAS: "2-D Curve Matching Using High Curvature Points: Application to Stereo Vision"

[Der90] R. DERICHE and G. GIRAUDON: "Accurate Corner Detection : An Analytical Study" In ICCV 90, Osaka Japan December 1990.

[Dre82] L. DRESCHLER and H.H. NAGEL: "On the Selection of Critical Points and local Curvature Extrema of Region Boundaries for Interframe Matching", in International Conference on Pattern Recognition, 1982, pp 542-544.

[Gui88] A. GUIDUCCI : "Corner Characterization by Differential Geometry Techniques", in Pattern Recognition Letters Vol 8 1988, pp 311-318.

[Har87] C.G. HARRIS: "Determination of ego-motion from matched points", Proc. Alvey Vision Conf. Cambridge, UK, 1987.

[Har88] C. HARRIS and M. STEPHENS :"A Combined Corner and Edge Detector", Proc 4th Alvey Vision Conf. Manchester August 1988 pp 189-192

[Hua86] H.Z. HUA and Y. QUIAN : "A direct corner detection algorithm", in Proc. 8th ICPR October 1986, Paris. pp 853-855

[Kit82] L. KITCHEN and A. ROSENFELD : "Gray-Level Corner Detection", in Pattern Recognition Letters, December 1982, pp 95-102.

[Kru89] W.M. KRUEGER and K. PHILLIPS : "The Geometry of Differential Operators with Application to Image Processing", in IEEE PAMI Vol11, December 1989 pp 1252-1264

[Lip69] M.M. LIPSCHUTZ : "Differential Geometry" McGraw-Hill, New York USA 1969

[Mar80] D. MARR and E. Hildreth: "Theory of Edge Detection", In Proc Roy Soc. London, B207, 1980 pp 187-217

[Med86] G. MEDIONI and Y. YASUMUTO : "Corner Detection and Curve Representation using cubic B-spline", IEEE Int. Conf, Robotics and Automation 1986, pp 764-769.

[Mic89] E. De MICHELI, B. CAPRILE, P. OTTONELLO, V. TORRE : "Localisation and Noise in Edge Detection", in IEEE PAMI, Vol 11, No 10, October 1989, pp 1106-1117.

[Mok86] F. MOKHTARIAN and A. MACKWORTH "Scale-based description and recognition of planar curves and 2D shapes", IEEE PAMI Vol8 No 1 1986 pp 34-43

[Mor77] H.P. MORAVEC : "Towards Automatic Visual Obstacle Avoidance" Proc Int. Joint Conf Artificial Intelligence Cambridge, MA, USA August 1977 pp 584

[Nag83] H.H. NAGEL: "Displacement Vectors Derived from Second-Order Intensity Variations in Image Sequences", in CVGIP Vol 21 1983, pp 85-117.

[Nob88] J.A. NOBLE : "Finding Corners", in Image and Vision Computing, Vol 6, may 1988 pp 121-128

[Ran89] K. RANGARAJAN, M. SHAH and D.V. BRACKLE : "Optimal Corner Detector", in CVGIP, Vol 48, 1989, pp 230-245

[Sha84] M.A. SHAH and R. JAIN: "Detecting Time-Varuing Corners", in CVGIP, Vol28 1984 pp 345-355.

[Sin90] A. SINGH and M. SHNEIER: "Grey Level Corner Detection: A Generalization and a Robust Real Time Implementation", in CVGIP Vol51, 1990, pp 54-69.

[Tor86] V. TORRE and T.A. POGGIO: "On Edge Detection", in IEEE PAMI, Vol8,No2, March 1986, pp 147-163.

8 Appendix

Differential Geometry for Surface Curvature Analysis

Differential geometry is an important tool for the analysis of surfaces. In this annex, we give some elementary results of surface differential geometry that are used in the paper. The differential properties given will be employed to perform detailed curvature analyses on surfaces and applied in order to get the exact position of the corner and trihedral vertex.

Let us consider the surface S(x,y) associated to the grey-level intensity image $I(x,y)$ described by the equation

$$\mathbf{S}(\mathbf{x},\mathbf{y}) = x\vec{i} + y\vec{j} + I(x,y)\vec{k} \tag{25}$$

An infinitesimal distance element between two neighboring points (x,y) and $(x+dx, y+dy)$ on this surface is given by :

$$\mathbf{dSdS} = \begin{bmatrix} dx & dy \end{bmatrix} \begin{bmatrix} E & F \\ F & G \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} \tag{26}$$

where the elements E,F and G of the defining matrix noted **G** are given by

$$\begin{aligned} E &= \mathbf{S_x S_x} \\ F &= \mathbf{S_x S_y} \\ G &= \mathbf{S_y S_y} \end{aligned} \tag{27}$$

This equation is known as the **first fundamental form** of the surface **S(x,y)** It is usually noted Φ_1 and can be rewritten as :

$$\Phi_1 = Edx^2 + 2Fdxdy + Gdx^2 \tag{28}$$

For an image surface described by equation (25), it is easy to show that the coefficients E,F and G reduce to :

$$\begin{aligned} E &= 1 + I_x I_x \\ F &= I_x I_y \\ G &= 1 + I_y I_y \end{aligned} \tag{29}$$

These metric coefficients provide the basis for the measurements of lengths and areas on the surface. The **first fundamental form** gives the distance $\mathbf{dS^2}$ between neighboring points (x,y) and $(x+dx, y+dy)$ on a surface, to first order in dx and dy.

[Zun83] O.A. ZUNIGA and R.M. HARALICK: " Corner detection using the facet model", in Proc. Conf. on Pattern Recognition Image Processing 1983 pp 30-37.

The distance element dS lies in the tangent plane of the surface at (x, y) and therefore yields no information on how the surface curves away from the tangent plane at the given point. To deal with surface curvature, we have to consider the displacement between neighboring points (x, y) and $(x + dx, y + dy)$ to second order in dx and dy. The component of this displacement perpendicular to the tangent plane at (x, y) is given by:

$$-\mathbf{dSdN} = \begin{bmatrix} dx & dy \end{bmatrix} \begin{bmatrix} L & M \\ M & N \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} \tag{30}$$

where **N** is the surface unit normal given by :

$$\mathbf{N} = \frac{S_x \wedge S_y}{\| S_x \wedge S_y \|} \tag{31}$$

and the elements L,M and N of the defining matrix noted **H** and known as the **Hessian matrix**, are given by :

$$\begin{aligned} L &= \mathbf{NS_{xx}} \\ M &= \mathbf{NS_{xy}} \\ N &= \mathbf{NS_{yy}} \end{aligned} \tag{32}$$

Equation (30) is known as **the second fundamental form** of the surface $\mathbf{S(x, y)}$. It is usually noted by Φ_2 and can be rewritten as :

$$\Phi_2 = L dx^2 + 2M dx dy + N dy^2 \tag{33}$$

The second fundamental coefficients L,M and N form the basis for defining and analyzing the curvature of a surface. For an image surface described by equation (25), it is easy to show that these coefficients reduce to :

$$\begin{aligned} L &= \frac{I_{xx}(x,y)}{\sqrt{1+I_x(x,y)^2+I_y(x,y)^2}} \\ M &= \frac{I_{xy}(x,y)}{\sqrt{1+I_x(x,y)^2+I_y(x,y)^2}} \\ N &= \frac{I_{yy}(x,y)}{\sqrt{1+I_x(x,y)^2+I_y(x,y)^2}} \end{aligned} \tag{34}$$

At each point of the surface, the family of planes containing the normal **N** at the given point cut the surface in a family of normal section curves for that point. The normal curvature κ of the surface at a given point is defined as the curvature of normal section curve at that point and is given by the ratio of the second and the first fundamental form:

$$\kappa = \frac{Ldx^2 + 2Mdxdy + Ndy^2}{Edx^2 + 2Fdxdy + Gdy^2} \tag{35}$$

At each point, a direction (dx, dy) is associated to the curvature κ given by the equation (35). There are two directions for which κ has maximum and minimum values. They are solution of the equations

$$\begin{aligned} \frac{\partial \kappa}{\partial (dx)} &= 0 \\ \frac{\partial \kappa}{\partial (dy)} &= 0 \end{aligned} \tag{36}$$

Using equation (35) and deriving yields the following equations:

$$\begin{aligned} (-L + \kappa E)dx + (-M + \kappa F)dy &= 0 \\ (-M + \kappa F)dx + (-N + \kappa G)dy &= 0 \end{aligned} \tag{37}$$

For a solution to exist, κ must satisfy

$$\kappa^2 - 2H\kappa + K = 0 \tag{38}$$

where the coefficients K and H denote the **Gaussian curvature** and **the mean curvature** respectively :

$$\begin{aligned} K &= \frac{LN - M^2}{EG - F^2} \\ H &= \frac{EN + GL - 2FM}{2(EG - F^2)} \end{aligned} \tag{39}$$

The solutions to equation (38) represent the maximum and minimum curvature at the given point and are called **the principal curvatures**

$$\begin{aligned} \kappa_{min} &= H - \sqrt{H^2 - K} \\ \kappa_{max} &= H + \sqrt{H^2 - K} \end{aligned} \tag{40}$$

Substituting κ_{min} and κ_{max} for κ in equation (38) yields the solutions (dx, dy) for the principal directions that are alway orthogonal.

When $H^2 = K$, $\kappa_{min} = \kappa_{max}$ and the curvature is independent of direction. Such point is called an *umbillic* or *spherical* point since the surface locally approximates a sphere at that point.

The **Gaussian** and **Mean Curvatures** introduced above can easily be expressed as the product and average of the **Principal Curvatures**, respectively :

$$K = \kappa_{max}\kappa_{min}$$
$$H = \frac{\kappa_{max}+\kappa_{min}}{2} \tag{41}$$

It is worthnoting that if we define the matrix β as follows :

$$\beta = [G^{-1}][H] \tag{42}$$

It can be shown that K and H can easily be expressed as the determinant and half the trace of the matrix β respectively.

More details and results about the differential geometry of surfaces can be found in the reference [Lip69].

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249 - 6399