Edge Detection Techniques - An Overview

Djemel Ziou*
Département de math et informatique
Université de Sherbrooke
Québec, Canada, J1K 2R1
Tel: (1) (819) 821 8000, ext: 2859
Fax: (1) (819) 821 8200
Djemel.Ziou@dmi.usherb.ca

Salvatore Tabbone
Crin - Cnrs/Inria Lorraine
54506 Vandoeuvre-Les-Nancy
France
(33) 3 83 59 30 39
(33) 3 83 41 30 79
Salvatore.Tabbone@loria.fr

Abstract

In computer vision and image processing, edge detection concerns the localization of significant variations of the grey level image and the identification of the physical phenomena that originated them. This information is very useful for applications in 3D reconstruction, motion, recognition, image enhancement and restoration, image registration, image compression, and so on. Usually, edge detection requires smoothing and differentiation of the image. Differentiation is an ill-conditioned problem and smoothing results in a loss of information. It is difficult to design a general edge detection algorithm which performs well in many contexts and captures the requirements of subsequent processing stages. Consequently, over the history of digital image processing a variety of edge detectors have been devised which differ in their mathematical and algorithmic properties. This paper is an account of the current state of our understanding of edge detection. We propose an overview of research in edge detection: edge definition, properties of detectors, the methodology of edge detection, the mutual influence between edges and detectors, and existing edge detectors and their implementation.

*Corresponding author

Contents

1 Introduction

In computer vision, edge detection is a process which attempts to capture the significant properties of objects in the image. These properties include discontinuities in the photometrical, geometrical and physical characteristics of objects. Such information give rise to variations in the grey level image; the most commonly used variations are discontinuities (step edges), local extrema (line edges), and 2D features formed where at least two edges meet (junctions).

The purpose of edge detection is to localize these variations and to identify the physical phenomena which produce them. Edge detection must be efficient and reliable because the validity, efficiency and possibility of the completion of subsequent processing stages rely on it. To fulfill this requirement, edge detection provides all significant information about the image. For this purpose, image derivatives are computed. However, differentiation of an image is an ill-posed problem; image derivatives are sensitive to various sources of noise, i.e., electronic, semantic, and discretization/quantification effects. To regularize the differentiation, the image must be smoothed. However, there are undesirable effects associated with smoothing, i.e., loss of information and displacement of prominent structures in the image plane. Furthermore, the properties of commonly-used differentiation operators are different and therefore they generate different edges. It is difficult to design a general edge detection algorithm which performs well in many contexts and captures the requirements of subsequent processing stages. Consequently, over the history of digital image processing a variety of edge detectors have been devised which differ in their purpose (i.e., the photometrical and geometrical properties of edges which they are able to extract) and their mathematical and algorithmic properties.

This paper describes the characteristics of edges, the properties of detectors, the methodology of edge detection, the mutual influence between them and the main idea behind the major edge detection techniques. In the next section, we will give a definition of edges and related physical phenomena. Section 3 is devoted to the properties of edge detectors. In section 4, we analyze the influence of image characteristics and the properties of an edge detector on its performance. Finally, we briefly describe existing edge detectors and their implementation in section 5.

2 Edge Definition

Physical edges provide important visual information since they correspond to discontinuities in the physical, photometrical and geometrical properties of scene objects. The principal physical edges correspond to significant variations in the reflectance, illumination, orientation, and depth of scene surfaces. Since image intensity is often proportional to scene radiance, physical edges are represented in the image by changes in the intensity function. The most common types of image intensity variations are steps, lines and junctions.

Steps are by far the most common type of edge encountered. This type of edge results from various phenomena: for example when one object hides another, or when there is a shadow on a surface. It generally occurs between two regions having almost constant, but different, grey levels.

The step edge is the point at which the grey level discontinuity occurs. In real images, step edges are localized at the inflection points of the image. In fact, the image formation process involves the convolution of the camera point-spread function with the edge profile corrupted by noise which produces a smooth function (see Fig 1.a and b). Consequently, step edges are localized as positive maxima or negative minima of the first-order derivative (Fig. 1.c) or as zero-crossings of the second-order derivative (Fig. 1.d). In 2D, the first derivative is defined by the gradient operator and the second derivative is approached by the Laplacian or by the second derivative along the gradient direction. This step edge definition does not include the spatial distribution of edges. It is more realistic to consider a step edge as a combination of several inflection points. The most commonly used edge model is the double step edge (two inflection points in the vicinity of each other). There are two types of double edges: the pulse (Fig. 1.e) and the staircase (Fig. 1.f). As we shall see below, the distinction between these two models is motivated by the fact that they produce different edge information when Laplacian detectors are used.

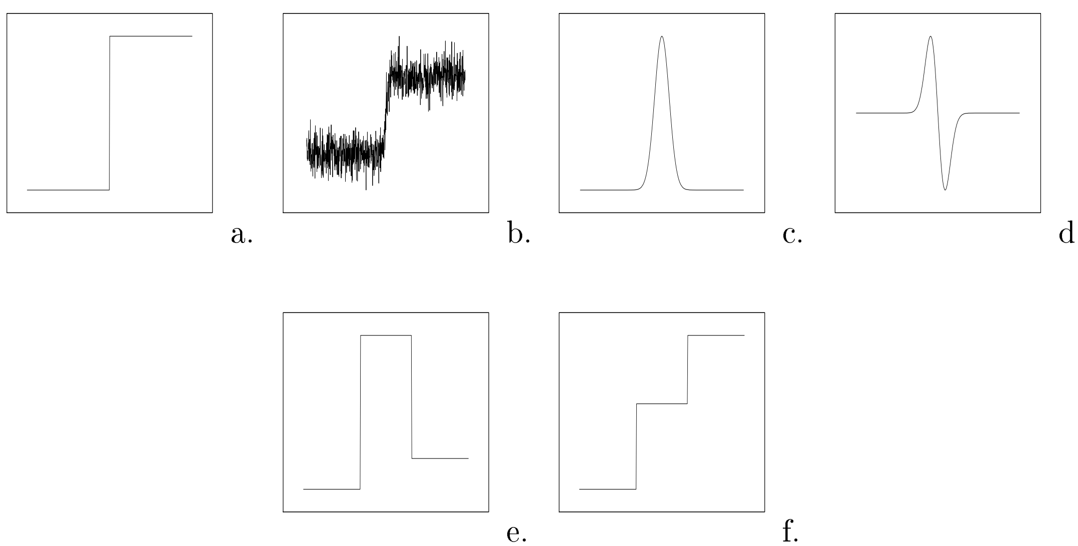


Figure 1: *Step edge profile. a) Ideal step edge, b) Smoothed step edge corrupted by noise, c) and d) First and second-order derivatives of the smoothed step edge (noise-free), e) Pulse, f) staircase.*

Lines result from mutual illumination between objects that are in contact or from thin objects placed against a background. The profile of lines is given in Figure 2.a. Lines correspond to local extrema of the image. They are localized as zero-crossings of the first derivative, or local maxima of the Laplacian, or local maxima of the grey level variance of the smoothed image. This type of edge is successfully used in remote sensing images for instance to detect roads and rivers [29].

In the scene, a physical corner or physical junction is formed when at least two physical edges meet. There are additional circumstances that may create corners in the image, for example illumination effects or the presence of occlusions e.g., a T-junction is created if one physical edge occludes another (see Fig. 2.b). In what follows we will use the terms "junction" and "corner" synonymously. In the image, a junction is a 2D-feature. It will be sharp and is defined as the point where two or more edges (which can be of different types) meet. There are several junction models: T, L, Y, X. The junction can be localized in various ways: e.g., a point with high curvature, or a point with great variation in gradient direction, or a zero-crossing of the Laplacian with high curvature or near an elliptic extremum.

The term "edge", as commonly used, encompasses all types of edges, but the majority of existing edge detection algorithms are adapted to step edges, which are the most common. Other types of edges are given in [87].

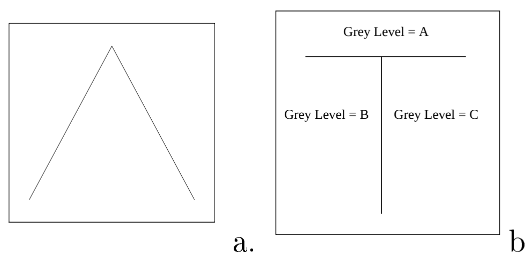
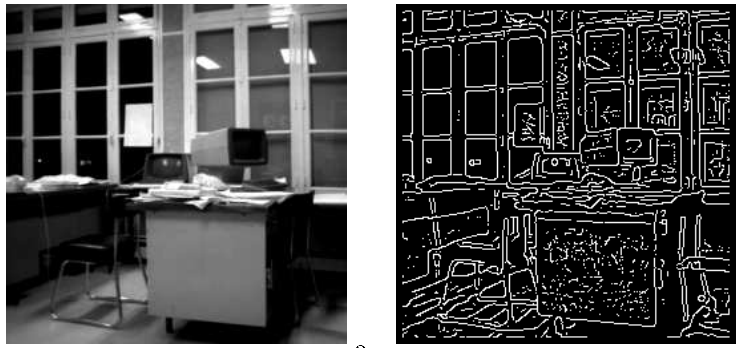


Figure 2: *a) Line profile, b) T-Junction profile.*

3 Properties of Edge Detectors

An edge detector accepts discrete, digitized images as input and produces an edge map as output. The edge map of some detectors includes explicit information about the position and strength of edges, their orientation, and the scale. The example in Figure 3 includes position information only.

During the history of image processing, a variety of edge detectors have been devised which differ in their purpose (the photometrical and geometrical properties of the edge) and in their mathematical and algorithmical properties. From the point of view of integration of an edge detector into a computer vision system there are two classes of detectors. The first includes detectors which do not use *a priori* knowledge about the scene and the edge to be detected. This



a. b.

Figure 3: *a) Original image, b) Position information provided by an edge detector.*

class of "autonomous" detectors is influenced neither by other components of the vision system nor by contextual information. These detectors are flexible in the sense that they are not limited to specific images. However, they are based on local processing; the process of labeling an edge is based only on its neighboring pixels. The second class of detectors are contextual, in the sense that they are guided by the results of other components of the system or by *a priori* knowledge about the edge or the structure of the scene. It follows that they perform only in a precise context. Few contextual detectors have been proposed. If we consider the knowledge used by the detectors, it is clear that autonomous detectors are appropriate for general-purpose vision systems. However, contextual detectors are adapted to specific applications where processed images always include the same objects.

Conceptually, the most commonly proposed schemes for edge detection (both autonomous and contextual detectors) include three operations: differentiation, smoothing and labeling. Differentiation consists in evaluating the desired derivatives of the image. Smoothing consists in reducing noise in the image and regularizing the numerical differentiation. Labeling involves localizing edges and increasing the signal-to-noise ratio of the edge image by suppressing false edges. Labeling is the last operation to be run. However, as will be shown below, the order in which differentiation and smoothing are run depends on their properties. Smoothing and differentiation of an image are realized only by filtering the image with the differentiation of the smoothing filter. In this regards, the terms filter and detector are often used synonymously [67, 41, 8, 21, 124]. The performance of these three operations is related. In fact, smoothing regularizes the differentiation and the specification of the false edge suppression step depends on the performance of the two other operations. If the smoothing step reduces noise without loss of information, false edge suppression is easy to accomplish.

The specification of an edge detector in terms of three operations is incomplete. In fact, an edge detector includes neither the precise context in which it can be successfully used nor the scale computation. It is necessary to define a methodology of edge detection to make explicit how to choose the scale (Multi-scale) and how to select an edge detector (multi-detector) for a target application. In what follows, we will present the three operations, the multi-detector and the multi-scale approaches in more details. The reader can find a study of smoothing filters and differentiation operators used in edge detection in the paper of Torre and Poggio [114].

3.1 Smoothing of the Image

It should be recalled that smoothing has a positive effect to reduce noise and to ensure robust edge detection; and a negative effect, information loss. Clearly, we have a fundamental trade-off here between loss of information and noise reduction. The ultimate goal is to find optimal detectors that ensure a favorable compromise between noise reduction and edge conservation. Regularization theory is the formalization, as an optimization problem, of the search for an optimal filter. In fact, image differentiation, like many other problems in early computational vision, is ill-posed in the sense that the existence, uniqueness, and stability of a solution cannot be guaranteed in the absence of additional constraints. Poggio [86] suggests that the class of admissible solutions to an ill-posed problem is restricted by imposing additional constraints. Usually, these constraints concern the appropriate compromise between noise reduction and edge conservation. For example, in standard Tikhonov's regularization, a problem is rendered well-posed by restricting the acceptable solutions to the space of smooth functions. In the edge detection context, Poggio and Torre [85, 86] show that regularizing differentiation can be accomplished by convolution of the image with the cubic spline (or its derivatives), with area controlled by a regularization parameter. The regularization parameter determines the compromise between noise elimination and the preservation of image structure. In filtering terminology, the regularization parameter is called the *scale*. In addition to the cubic spline, two other regularization filters have been proposed in [86], the Green function and the Gaussian, whose respective impulse responses are:

$$r(x) = \frac{\mu}{2\sqrt{2}} e^{-\mu|x|} (cos(\frac{\mu|x|}{\sqrt{2}}) + sin(\frac{\mu|x|}{\sqrt{2}})) \tag{1}$$

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

where $\sigma > 0$ and $\mu > 0$. Although these filters ensure a compromise between noise elimination and the preservation of image structure, we are faced with the problem of choosing the regularization parameter. This is important since judicious selection of this parameter reduces information loss. This problem will be presented below (cf. 3.4).

The attributes of a smoothing filter that influence the performance of the edge detector are its linearity, the duration of its impulse response, and its invariance to rotation. Non-linear filtering has been proven to be more successful than linear filtering because it removes certain kinds of noise better (e.g., impulse noise) while preserving edge information [84]. However, in this section we will focus on linear filtering because it is more common in edge detection. The duration of the impulse response characterizes the support of the filter in the spatial or frequential domain. For instance, in edge detection, three kinds of linear low-pass filters have been used: band-limited filters, support-limited filters and filters with minimal uncertainty. The invariance to rotation property ensures that the effect of smoothing is the same regardless of edge orientation.

A sufficient condition for a function or operator (functional) to be rotationally invariant is that its polar form depends only on the radial distance $\rho = \sqrt{x^2 + y^2}$ and not on its direction $tan(\phi) = y/x$. Formally, a function $f(x, y)$ is said to be invariant to rotation (we may also call it a radial or rotationally symmetric function) iff for all ϕ, $f(x, y) = f(X, Y)$, where $X = xcos(\phi) + ysin(\phi)$ and $Y = -xsin(\phi) + ycos(\phi)$. For example, the Gaussian is invariant to rotation, whereas the 2D version of the filter defined in (eq. 1) is not. This property will be examined in more detail in section 4.2. Other information on this topic can be found in [6, 15, 125, 129].

3.2 Image Differentiation

It should be recalled that the purpose of edge detection is to localize variations of the image grey level and to identify the physical phenomena which produced them. Differentiation is the computation of the necessary derivatives to localize these edges. the differentiation operator is characterized by its order, its invariance to rotation and its linearity.

The order of the differentiation operator is defined by the order of its partial derivatives. An operator $\mathcal{O}_{x,y}$ is invariant to rotation iff $\mathcal{O}_{X,Y} = R\mathcal{O}_{x,y}$, where R is the rotation matrix. $\mathcal{O}_{x,y}$ is linear iff for all positive scalars α and β and for all functions $f(x, y)$ and $g(x, y)$, we have $\mathcal{O}_{x,y}(\alpha f(x, y) + \beta g(x, y)) = \alpha\mathcal{O}_{x,y}(f(x, y)) + \beta\mathcal{O}_{x,y}(g(x, y))$. The most commonly used operators are the gradient, the Laplacian and the second-order directional derivative. As we will show, the properties of the operator to be used are determined by the characteristics of the image and the subsequent use of edges.

The gradient is a first-order operator defined as the vector $(\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$. The modulus and the direction of the gradient are defined by:

$$|\vec{grad}| = \sqrt{(\frac{\partial}{\partial x})^2 + (\frac{\partial}{\partial y})^2} \qquad \text{and} \qquad \psi = arctg\left(\frac{\partial}{\partial y} / \frac{\partial}{\partial x}\right) \tag{3}$$

The gradient direction is perpendicular to the edge orientation. In many proposed schemes, the gradient direction is used to localize edges. The gradient modulus operator is non-linear and invariant to rotation. It is only computed by using derivatives in x and y. In a noisy image, the use of several directional derivatives may be useful for increasing the signal-to-noise ratio. From a computational viewpoint, according to the steering theorem [32], derivatives of the image in any direction can be expressed as the weighted sum of the derivatives of the image in particular directions. It should be noted that due to the computational inefficiency of the square-root operation required in (eq. 3), the gradient modulus is often calculated using one of two other metrics: $|\frac{\partial}{\partial x}| + |\frac{\partial}{\partial y}|$ and $max(|\frac{\partial}{\partial x}|, |\frac{\partial}{\partial y}|)$. While these can be calculated more efficiently, it is pointed out in [66] that in the case of a step edge the accuracy of the estimated gradient modulus is reduced.

The second-order operators are defined in terms of $\partial^2/\partial x^2$, $\partial^2/\partial x \partial y$ and $\partial^2/\partial y^2$. The commonly used operators in edge detection are the Laplacian and the second-order directional derivative along the gradient direction. These operators are defined by:

$$\nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad \text{and} \quad \frac{\partial^2}{\partial \vec{n}^2} = \frac{\partial^2}{\partial x^2} cos^2(\psi) + \frac{\partial^2}{\partial x \partial y} sin(\psi) cos(\psi) + \frac{\partial^2}{\partial y^2} sin^2(\psi) \tag{4}$$

where $\vec{n}$ is the gradient direction. The Laplacian operator is linear and rotationally symmetric, whereas the second directional derivative is neither linear nor invariant to rotation.

Let us now examine the order in which smoothing and differentiation are performed. The appropriate sequence of these operations depends on the linearity of the differentiation operator [114, 122]. Linear operators are associative and commutative with convolution. Formally, for $I(x, y)$ and $f(x, y)$, two L^2 functions differentiable according to the linear operator $\mathcal{O}_{x,y}$, we have the following result:

$$\mathcal{O}_{x,y}(I(x, y) * f(x, y)) = \mathcal{O}_{x,y}(I(x, y)) * f(x, y) = I(x, y) * \mathcal{O}_{x,y}(f(x, y)) \tag{5}$$

Consequently, the order in which smoothing and differentiation are performed is immaterial, since they are commutative. Non-linear operators are neither associative nor commutative with convolution. The regularization requirement (cf. 3.1) implies in this case that smoothing must precede the differentiation operation.

3.3 Edge Labeling

Edge labeling involves localizing edges and increasing signal-to-noise ratio by suppressing false edges. The localization procedure depends on the differentiation operator used. In the early gradient detectors, edges were localized by thresholding the gradient modulus. The edges which resulted from this method where not filiform and consequently a skeletization operation was required. An improvement has been achieved by the use of the non-maximum suppression algorithm. The basic idea is to extract local maxima of the gradient modulus. An efficient algorithm has been proposed in [7] which finds the local maxima along the direction of the gradient vector. That is, if we consider the image plane as real, then a given pixel is a local maximum if the gradient modulus at this pixel is greater than the gradient modulus of two neighboring points situated at the same distance on either side of the given pixel along the gradient direction. For second-order detectors the localization of zero-crossings is as follows: the output of a second-order detector at a given pixel is compared with neighbor pixels to the left and below it. If these three pixels do not have the same signs, there is a zero-crossing. However, it is shown in [108] that the use of more than the two principal directions (horizontal and vertical) improves the localization, especially for certain junction models (e.g., L-junction). Another well-known scheme is described in [44]. The output image is inspected to see whether it matches one of eleven allowable zero-crossing predicates.

The elimination of false edges increases the signal-to-noise ratio of the differentiation and smoothing operations. In spite of the importance of the cleaning operation, little works have been done on this subject. While it may be true that the behavior of this operation depends on the performance of smoothing and differentiation and that these operations are more and more robust, false edges do not originate only from noise. As explained below, there are other phenomena that give rise to false edges. The rule commonly used to classify edges as true or false is that the plausibility value of true (resp. false) edges is above (resp. below) a given threshold. The threshold is the minimum acceptable plausibility value. Due to fluctuation of the plausibility measure, edges resulting from such a binary decision rule are broken. So this rule has been improved by using the hysteresis algorithm [8] to take into account edge continuity. Two thresholds are used; a given edge (e.g., an ordered list of edge points) is true if the plausibility value of every edge point on the list is above a low threshold and at least one is above a high threshold. Otherwise, the edge is false.

Specification of the cleaning process depends on the differentiation operator used. When the gradient differentiation operator is used, the plausibility measure is the gradient modulus. False maxima originate from noise and can be eliminated using one of the above-mentioned rules. One problem of image cleaning concerns the choice of the plausibility measure when a second-order differentiation operator is used. Many authors [100, 13] use the gradient modulus as a plausibility measure. However, it is well known that an authentic zero-crossing can correspond to a weak gradient modulus (i.e., saddle point). Other authors [67, 41, 108, 109, 127] suggest the use of the slope of the edge as a plausibility measure. However, computation of the slope requires the computation of third-order derivatives and hence it is noise-sensitive. False zero-crossings have at least two causes and require the use of an appropriate thresholding method [12, 13, 108, 109, 127]. The first class of false edges is well known: originating from noise, they are called "noised edges". The noise results from both the image acquisition system and the nature of the scene under consideration (i.e., texture). In many images, this type of false edge usually has a low gradient modulus and can be discarded using an algorithm such as hysteresis. The second class of false edges, called "phantom edges", arise from certain edge models (e.g, staircase).

Intuitively, they are zero-crossings of the second derivative which correspond to the positive minima or negative maxima of the first derivative of the staircase (see Figs. 4 and 5). It is pointed out in [13] that for the case of white, normally-distributed random signals, the phantom edges are fewer (by a factor of nine) and weaker (by a factor of about three) than the authentic edges. In practice, however there may be phantom edges whose gradient magnitude is greater than that of some authentic edges. In addition, a phantom edge usually forms a continuous curve which extends an authentic curved edge. Therefore, the use of edge continuity as a criterion to clean an image is not appropriate, since it implies the non-suppression of phantom edges. Phantom edges as minima of the gradient magnitude are the only edge points which verify the following condition. Let $I(x, y)$ be the input image and $f(x, y)$ the smoothing filter. An edge (x_0, y_0) is false if and only if:

$$\frac{\partial^2 (I*f)'_{\vec{n}}(x_0,y_0)}{\partial x^2}\frac{\partial^2 (I*f)'_{\vec{n}}(x_0,y_0)}{\partial y^2} - \left(\frac{\partial^2 (I*f)'_{\vec{n}}(x_0,y_0)}{\partial x \partial y}\right)^2 > 0 \quad \text{and} \quad \frac{\partial^2 (I*f)'_{\vec{n}})(x_0,y_0)}{\partial x^2} > 0 \tag{6}$$

where $(I * f)'_{\vec{n}}(x, y)$ is the first derivative of $(I * f)(x, y)$ taken along the direction of the gradient. Given the original image and the corresponding zero-crossings, the cleaning algorithm consists of two steps [13, 108, 109, 127]: 1) Elimination of phantom edges by using the condition given in (eq. 6), 2) Elimination of noisy edges by using the hysteresis algorithm (or any other thresholding algorithm).

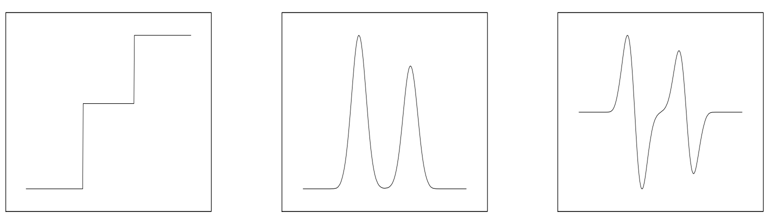


Figure 4: *a) The profile of the staircase edge, b) Two maxima and one minimum of the first derivative of the smoothed staircase, c) Three zero-crossings of the second derivative of the smoothed staircase.*

Another aspect of the elimination of false edges concerns threshold computation. Usually, a threshold value is found using a trial-and-error process and is used for all edges of an image. However, it is pointed out in [18, 111, 127, 128] that the threshold is a function of edge characteristics, properties of the smoothing filter, and properties of the differentiation operator. Consequently, it is not easy to find a single threshold value for a given image. An automatic rule to compute the threshold for the Laplacian of Gaussian detector has been proposed in [18]. This rule is empirical, no justification has been given and it has been tested only on synthetic data. We have proposed in [108, 109, 111, 127] a cleaning rule for multi-scale edge detection based on the behavior of the ideal step edge in scale space. The threshold is found at a high scale and propagated automatically to ideal step edges obtained at lower scales. Improvements of this algorithm are proposed in [128] for use with any smoothing filter, differentiation operator, and edge model.

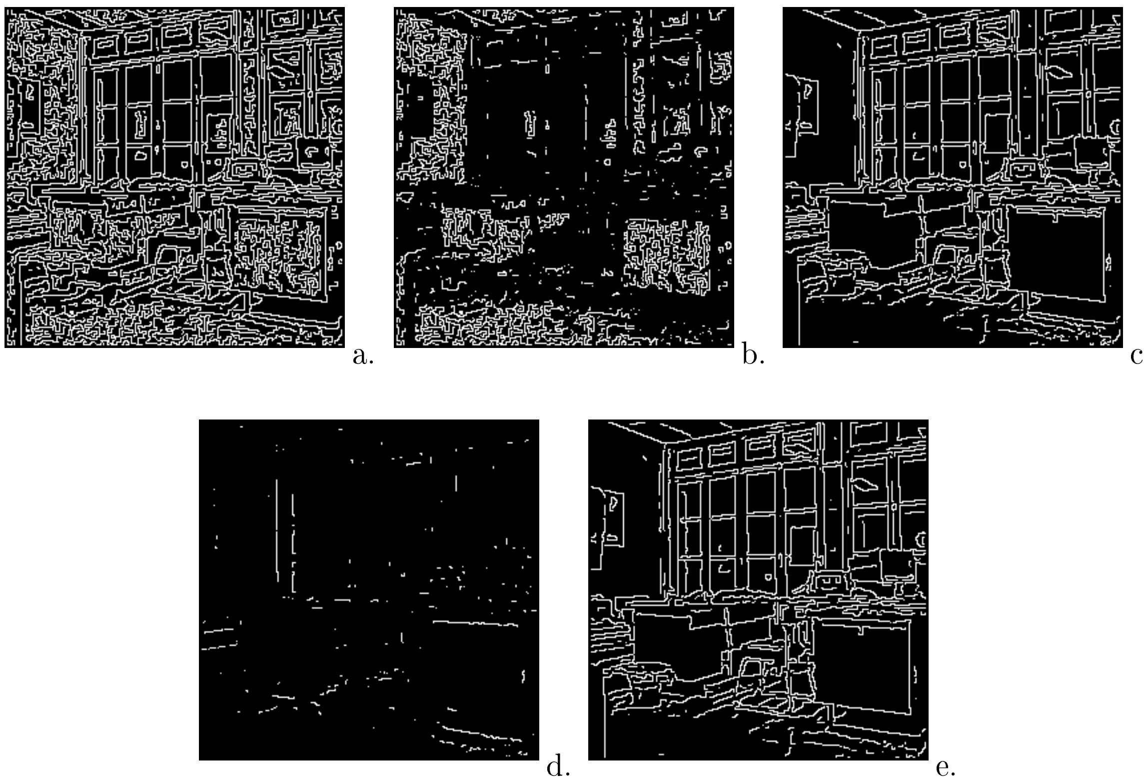
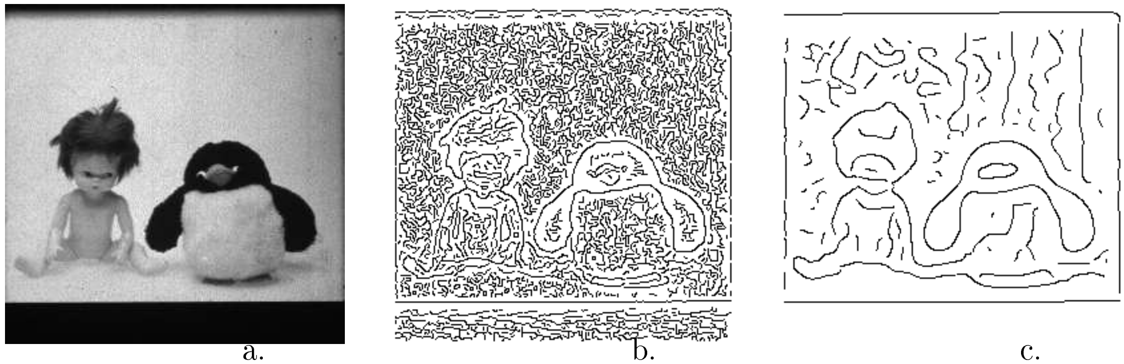


Figure 5: *a) Zero-crossings of the Laplacian of Gaussian with* $\sigma = 1$*, b) False edges caused by noise, c) Zero-crossings without false edges due to noise, d) False edges generated by the staircase of the image 5.c, e) Zero-crossings without false edges due to staircase or noise.*

3.4 Multi-Detector and Multi-Scale Approaches

In general, an edge detector includes neither the precise context (*i.e*, characteristics of the image) in which it can be successfully used nor the scale computation rule. For instance, certain detectors are commonly viewed as a convolution operation $(f_s * I)(x, y)$ where I is the image, f_s the filter and s the scale. In computer vision, we are often confronted with the problem of selecting an appropriate edge detector. Usually, the approach used consists of arbitrarily choosing a detector and using it to find all the edges in the images being processed in the target application. The scale is often fixed by trial-and-error experiments and reused for all images. It is obvious that this approach does not lead to correct results. In fact, one detector running at one scale does not yield all edges of the image [126]. It is more suitable to focus on edges by using several detectors that differ in their scales, mathematical properties and goals. The underlying problems are multiple; various knowledge and know-how about image formation and processing techniques are required to arrive at an effective approach. Matsuyama [68] pointed out the problems frequently encountered in designing image analysis systems. A small step in this direction has been accomplished by using multiple detectors and multiple scales as described in [38, 126]. Hasegawa *et al.* [38] have implemented the IMPRESS system that is able to choose the appropriate detector to find a given edge. It involves applying all the detectors and retaining the one which generates the result most similar to the reference edge. Ziou and Koukam [126] have implemented the SED system which is able to automatically select edge detectors and their scales to extract a given edge. To avoid a combinatorial approach, the selection criteria combine several sources of information such as edge characteristics, detector properties, the mutual influence between edges and detectors, and the effective results of the run detectors.

Multi-scale edge detection is a particular solution to this problem since it is limited to the use of single edge detector with multiple scales. Let us consider the gradient of the Gaussian. For a small scale, this detector is rather noise-sensitive; some extracted edges are twisted and broken but the fine details of intensity changes are obtained; for a large scale, coarse intensity changes are obtained but some edges have a large delocalization error (see Fig. 6). It is difficult to find a single scale which leads to an optimal detector for all edges in an image. As suggested by Rosenfeld and Turson [94] as well as Marr and Hildreth [67], we can obtain a description of an image at different scales by applying an edge detector at different scales and combining the recovered edge information. This is called the multi-scale approach. Let us examine the order in which smoothing and differentiation can be performed. We previously (cf. 3.1) gave a response for this question in the case of a detector which runs at one scale. In the case of multi-scale edge detection the order in which these operations run is slightly different. Traditionally, in multi-scale edge detection schemes, images are smoothed at several scales after differentiation. Let us consider two requirements of multi-scale edge detection: regularization and nice scaling behavior. The regularization requirement implies the following order: smoothing, differentiation and successive smoothing at different scales. Nice scaling behavior means that edges are not created as the scale increases; in general this requirement does not hold for non-linear operators [121, 122]. Many schemes for multi-scale edge detection have been proposed [118, 4, 82, 64, 127]. The use of this approach involves two problems: (1) the number of scales of an edge detector which can be used for a given image and the way in which those scales are selected, (2) the way in which the edge information recovered at different scales can be efficiently combined. It should be recalled that the scale is related to the regularization parameter (cf. 3.1).



a. b. c.

Figure 6: *a) Original image, b) and c) Gradient Maxima obtained at $\sigma = 1$ and $\sigma = 4$, respectively.*

As we just have seen, an edge detector is usually run at different scales; $s_1, s_1 + \Delta s, s_1 + 2\Delta s, \cdots, s_2$. When Δs is negative, this means that coarse edges are extracted before fine edges, thus the appellation coarse-to-fine strategy; whereas positive Δs corresponds to fine-to-coarse strategy. The choice of s_1, s_2, and Δs depends on the image characteristics, the properties of the detector used and the subsequent use of edges. But also Δs must be chosen so that the movement of edges obtained at consecutive scales in scale space will be small, thus making it easier to combine them. In his study, Bergholm [4] suggests $\Delta s = 0.5$ in the gradient of Gaussian case. This manner of choosing scales lacks efficiency since the scale is increased or decreased by the same amount (i.e., Δs) at each run of the detector. However, certain edges are invariant in scale space and do not need to be processed at each scale. This redundancy problem can be circumvented if it is possible to predict the scales that causes an edge to change. Various schemes for the automatic estimation of scales have been proposed [45, 34, 65, 52, 27]. The integration of these schemes in multi-scale edge detection schemes seems to be promising.

The Edge combination process involves identically labeling, with minimum error, edges obtained at different scales which originated from the same physical phenomena. The similar edges are combined to form a single image, to which are added all edges which are not matched. The resulting image is more complete and includes edges with minimal redundancy. Generally, the labeling process is not an easy one, edge combination can be made easier if the detector used has nice properties in scale space: no edges are created as the scale increases. When this property is fulfilled all edges appear at the finer scale and therefore edges can be tracked in scale space; for instance in fine-to-coarse tracking edges that disappear never reappear. In other words, this property provides a means of relating the descriptions at different scales to one another and allows the characterization of edges behavior in scale space. This behavior is an additional source of knowledge which can be used in combining edges and selecting scales. Theoretical studies [122, 3, 121] show that in one dimension, with the second derivative, the Gaussian filter is the only filter in a broad class which never creates zero-crossings as the scale increases. In two dimensions, this property is fulfilled by the Laplacian of Gaussian. It should be noted that in practice the consequences of the nice behavior property are not clear. The nice behavior of an edge detector also depends on the normalization constant that reduces approximation errors [104, 31, 44]. For example, certain authors [41, 44] consider the Laplacian of Gaussian as $K(1-\frac{\rho^2}{\sigma^2})e^{-\frac{\rho^2}{2\sigma^2}}$, where K is the normalization factor and $\rho^2 = x^2+y^2$. The value of the constant K is chosen so that the detector has a certain behavior in scale space, the sum of the discrete filter coefficients is zero and they are represented to the desired number of bits. Recently, Williams and Shah [119] suggest the use of the normalized version of the Gaussian; for a one-dimensional first-order derivative, this means that the area between the curve and the x-axis is in two parts; that for $x < 0$ is above the axis, while that for $x > 0$ is below. The total area of both regions under the curve is constant and equal to one. The normalized version of the Gaussian ensures that the gradient magnitude of the edges increases and decreases as a function of edge profile and edge interaction. For instance, if the edge is an ideal step, its first derivative does not depend on the scale.

Witkin [120] was one of the first to explicitly propose a description of a one-dimensional signal by its zero-crossings across scales. The signal is smoothed by a Gaussian filter using various scales and the zeros of the second derivative are located (see Fig. 7). In this description, called scale space, each zero-crossing is represented by its location at a fine scale and the scale at which it disappears. Witkin's description has allowed the study of the behavior of edge models as a function of the scale [119, 118, 4, 82, 30, 60, 87, 63, 127] and thus the use of the knowledge resulting from these studies to design edge combination algorithms. Canny [7, 8] has proposed a fine-to-coarse combination strategy of edges resulting from the use of his detectors at different scales. The combination process, called feature synthesis, involves using its detector to mark edges at a fine scale. From these edges, the large gradient output is synthesized and compared to the actual detector output. Additional edges are marked only if the large detector has significantly greater response than that predicted from the synthetic response. This combination process picks up fine details of the image and provides a smearing edges. Bergholm [4] proposes an algorithm, called edge focusing, for combining edge information, moving from a coarse to a fine scale. Canny's detector is run at decreasing scales in the vicinity of the recovered edges and the combination process consists of refining the previously recovered edges by the current ones. A drawback of Bergholm's algorithm is that some edges (i.e., the blurred ones) present a juggling phenomenon at small scales. Improvements of Bergholm's scheme have been proposed by Williams and Shah [118]. These authors consider Canny's detector and a general double edge model with different steps. They study the behavior of these edges in scale space and provide an equation for the movement of edges as a function of the scale. This equation describes the position of an edge at a given scale and can thus be used in the edge linking process. The schemes of Bergholm and Williams *et al.* are based on reasoning in scale space. They result from a thorough study of the behavior of edge models (step edge, double edge, corner edge, and box edge) in scale space. Lacroix [56], on other hand, uses a fine-to-coarse analysis to avoid this problem. However, the resulting edges have a large delocalization error at high scales. To avoid these problems Lindeberg [62] proposes labeling edges when they have a maximum gradient magnitude in scale space. All of these schemes use the gradient of the Gaussian. As mentioned above, only the Laplacian of Gaussian has the nice behavior. Lu and Jain [63, 64] consider different types of step edges and derive thirty-five rules describing the behavior of zero-crossings of the Laplacian of Gaussian in scale space. These rules allow multi-scale reasoning for the automatic selection of scales, edge localization, edge fusion, correction of edge location, and elimination of false edges. Tabbone and Ziou [127, 111] use neither the coarse-to-fine nor the fine-to-coarse strategy to combine edge information. The starting point for image description is the behavior of four step edge models (ideal, blurred, pulse, and staircase) in scale space. It is shown that, for these models, use of two scales (high and low) leads to a complete, correct description of the image in terms of zero-crossings. The authors propose rules for the combination of zero-crossings obtained at two scales: step and double edges are recovered at a low scale and blurred edges at a high scale. The smoothing filter used in this scheme is not the Gaussian and therefore the nice behavior is not fulfilled.

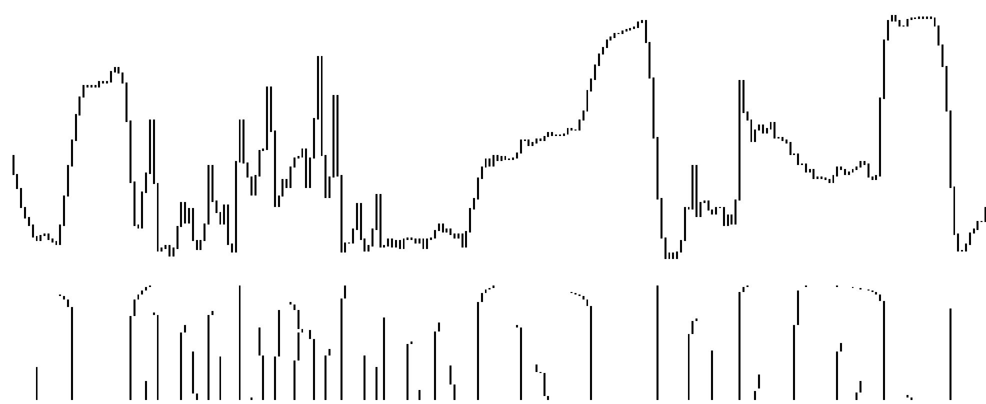


Figure 7: *One-dimensional signal smoothed at different scales and its Scale space representation.*

4 Mutual Influence Between Detectors and Edges

So far, we described the fundamental operations of an edge detector, i.e., smoothing, differentiation and labeling. In this section, we will deal with how image characteristics and the properties of smoothing and differentiation influence the computed edges. It should be recalled that an edge detector can be seen as a convolution operation and that it includes neither the precise context in which it can be successfully used nor the scale computation. The mutual influence relation between edges and detectors makes it possible to specify the context in which it is meaningful [126]. For example, the choice of a specific edge detector for a given image requires knowledge of its performance. Thus, this mutual influence relation leads to a better use of edge detectors and resulting in a better performance. In what follows, we will consider two of the three criteria of Canny as the performance of an edge detector: the signal/noise ratio (non-detection of true edges and detection of false edges) and the delocalization error (the distance between the location of the true edge and the location of the edge detected). The multiple response criterion do not concern edges, but only image noise; it is not considered here.

In the next section, we will present the influence of an edge detector's properties on the detected edges. We will discuss the effect of image characteristics, particularly subpixel and edge orientation. Finally, we will present the evaluation schemes used to measure the influence between edges and detectors.

4.1 Influence of an Edge Detector

It should be recalled that we are interested in the properties of the smoothing and differentiation operations. In this section, we will present the influence of the smoothing filter, especially the effect of its impulse response duration and the influence of the differentiation operator.

Let us examine informally the performance of the three kinds of low-pass filters commonly used: band-limited filters, support-limited filters and filters with minimal uncertainty. Band-limited filters are an obvious choice for regularizing differentiation, since the simplest way to avoid noise is to filter out high frequencies that are amplified by differentiation. However, their support is theoretically infinite and they are truncated before their implementation in the spatial domain. Consequently, the properties of the original filter are not conserved and the results obtained may not fulfill the initial specification. Band-limited filters are localized in the frequency domain to reduce the range of scales over which intensity changes take place. An example of a detector based on this kind of filter is proposed in [98]. Support-limited filters are not good for regularization. Differentiation reintroduces as many high frequencies as are removed by this type of filter. However, in contrast to band-limited filters, support-limited filters can be implemented in the spatial domain without truncation. The properties of the implemented filter are similar to those of the original filter. Support-limited filters are localized in the spatial domain (the filtered image should arise from a smooth average of nearby points). The difference-of-boxes filter [40] is an example of a support-limited filter. Filters with minimal uncertainty regularize the differentiation and they are an optimal compromise between the two previous filters. For example, the Gaussian filter (eq. 2) provides an optimal trade-off between these conflicting requirements.

Now, we will deal with the influence of the differentiation operators. We will examine the characteristics of edges produced by commonly-used differentiation operators and discuss conditions under which these operators give similar edges. The operators considered are the gradient, the Laplacian and the second-order derivative along the direction of the gradient.

We will begin by establishing the equivalence of the gradient and the second derivative along the gradient direction. The derivative of the gradient magnitude $M(x, y)$ along the gradient direction $\vec{n}$ is equal to the second-order derivative along the direction of the gradient.

$$\frac{\partial M(x,y)}{\partial \vec{n}} = \frac{\partial^2}{\partial x^2} cos^2(\psi) + \frac{\partial^2}{\partial x \partial y} sin(\psi)cos(\psi) + \frac{\partial^2}{\partial y^2} sin^2(\psi) \tag{7}$$

The equivalence between these operators is deduced from this equation. To clarify, we present the following example. Let us consider a vertical double edge $i(x, y) = erf(\frac{x-a}{s}) + erf(\frac{x+a}{s})$, where a and s are positive real numbers. For sake of simplicity and without loss of generality, we will use the notation $i(x)$ instead of $i(x, y)$, since the variable y in the right-hand term of $i(x, y)$ is not used. The derivatives of this signal at the origin are $i^{'}(0) = \frac{4}{s\sqrt{\pi}} e^{-a^2/s^2}$, $i^{''}(0) = 0$, and $i^{'''}(0) = \frac{8}{s^5\sqrt{\pi}} e^{-a^2/s^2}(2a^2 - s^2)$. The second derivative is zero at the origin. Under the condition $2a^2 > s^2$, this zero-crossing corresponds to a positive minimum of $i^{'}(x)$. In other words, for certain scales (i.e., $s < a\sqrt{2}$), if the first-order derivative is used, no edge should be labeled at the origin. However, if the second derivative is used an edge should be labeled at the origin. Even in a noise-free signal, these two operators give the same edges as output in the case of an isolated step edge. However, they give different results in the case of several edges which are close to each other (i.e., double edges). The second derivative gives more edge points than the first derivative. This difference is accentuated in the presence of noise. We conclude that there is an equivalence between these operators in the case of noise-free isolated step edges. However, in practice the equivalence is meaningless.

The Laplacian and the second-order derivative along the direction of the gradient are both second-order operators. They are related by:

$$\nabla = \frac{\partial M(x,y)}{\partial \vec{n}} + M(x,y)\frac{\partial \theta(x,y)}{\partial \vec{n_{\perp}}} \tag{8}$$

where $\vec{n_{\perp}}$ is the unit vector normal to the gradient direction. The Laplacian is decomposed into the change of magnitude of the gradient vector along the direction of the gradient and the change of direction of the gradient vector, multiplied by the gradient magnitude, along the direction perpendicular to the gradient. The term $\frac{\partial \theta(x,y)}{\partial \vec{n_{\perp}}}$ is related to the curvature of the underlying edge. Curvature is $\frac{\partial \theta(x,y)}{\partial s}$, where s is the arc length. Because $\vec{n_{\perp}}$ is aligned along the edge and closely approximates s, $\frac{\partial \theta(x,y)}{\partial \vec{n_{\perp}}}$ approximates the curvature. In particular, $\frac{\partial \theta(x,y)}{\partial \vec{n_{\perp}}}$ is zero only if the underlying edge is straight and theoretically infinite at the junction points. Given an image $I(x, y)$, the operators ∇ and $\frac{\partial M(x,y)}{\partial \vec{n}}$ are equivalent; that is their zero-crossings coincide, iff the curvature of the edge is zero, i.e., $\frac{\partial \theta(x,y)}{\partial \vec{n_{\perp}}} = 0$. This result is similar to the one in [114] and generalizes the result proposed in Marr's paper [67], i.e, $\nabla I(0, y) = \frac{\partial^2 I(0,y)}{\partial x^2} = 0$ iff the image $I(0, y)$ is constant or linear. Furthermore, the zero-crossings of the Laplacian have geometrical properties. Torre and Poggio [114] show that zero-crossing contours are closed curves or curves that terminate at the boundary of the image. However, edges obtained with directional operators do not have special geometrical properties.

Differentiation operators have been used to extract junctions. In this context, their performance has been studied in [5, 19, 78, 112]. It is shown that the commonly-used curvature measure is suitable for the detection of L-junctions but unpredictable for other junction models. Concerning the Laplacian operator, it is shown in [112] that for linear junction models (i.e., L, X, Y) with infinite extent and constant illumination (the edges forming the junction are linear, with infinite extent, and the underlying surfaces of the junction have a constant illumination) the Laplacian of Gaussian is zero at the junction point, while it is not zero for the other models, i.e., linear models with non-constant illumination, non-linear models, and models with finite extent. In the case of linear models with non-constant illumination, the value of the Laplacian of Gaussian depends on the image intensity variation. For non-linear models the value of the Laplacian of Gaussian depends on the model curvature. For finite extent linear models with constant illumination the Laplacian of Gaussian value tends to zero when the area of the model is large compared to the scale of the Gaussian. It is clear that in many cases, true junctions do not coincide with zeros of the Laplacian of Gaussian. We conclude that approaches that use zero-crossings of the Laplacian of Gaussian are not precise.

The performance of the gradient of the Gaussian have also been studied in [19]. The gradient scheme does not produce false edges in the vicinity of T and Y, as with the Laplacian. However, for the Y, T, V, and L linear models with infinite extent and constant illumination there is no local maximum of the gradient modulus along the direction of the gradient at the junction point. Moreover, the delocalization error of both maxima of the gradient and zeros of the Laplacian is large in the vicinity of the Y, T, V and L junctions (see Fig. 8).

4.2 Influence of Edge Characteristics

According to the definition given above, an edge is represented by its geometric and photometric characteristics. We are interested in those characteristics that influence the performance of an edge detector. The geometric characteristics are position, orientation, and smoothness. The photometric characteristics are an accurate description of the detailed variation of image intensity in the vicinity of the edge. We will consider two kinds of edge profile: the isolated step edge and the double edge. Let us assume that the surfaces of the image are linear. This is a reasonable assumption if the image is smoothed before edge analysis. The attributes of a step edge are its noise, contrast (the cumulative intensity change that occurs across the edge), steepness (the surface slope within the interval, across the profile, in which the bulk of the

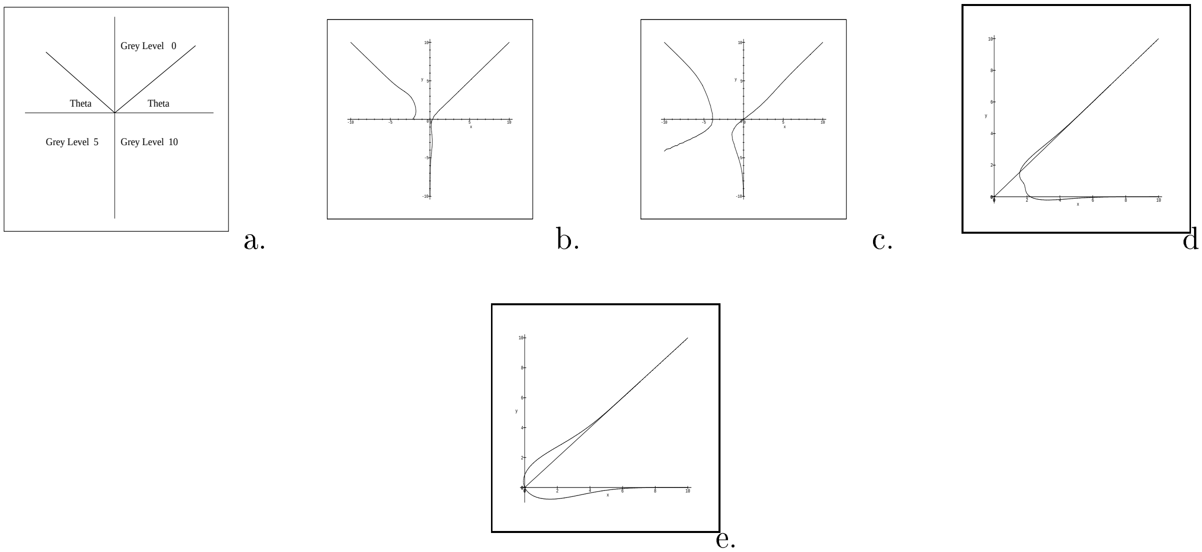


Figure 8: *a) Y-junction model, b) Y-junction edges obtained as maxima of the gradient, c) Y-junction edges obtained as zero-crossings of the Laplacian, d) L-corner edges obtained as maxima of the gradient, e) L-corner edges obtained as zero-crossings of the Laplacian.*

intensity change occurs), and finally its width (the size of this interval). The attributes of a double edge are the attributes of each step plus the distance between these two steps. The influence of edge characteristics on the performance of detectors has been studied by many investigators [1, 49, 48, 115, 19, 66]. We will give some knowledge which is independent of a particular detector. We will start with a brief description of the influence of photometric characteristics such as noise, steepness and edge type. Then, we will present the influence of geometric characteristics such as smoothness, subpixel, and orientation.

For the influence of noise, the higher the noise energy, the lower the signal/noise ratio, the greater the delocalization error is. In the case of a single step edge, the lower the edge steepness, the lower the signal/noise ratio and the greater the delocalization error. The delocalization error is greater in the case of a double edge because of the mutual influence between its two steps. It increases with the quotient of the detector scale and the distance between the two steps.

With regards to the influence of form attributes and smoothness, it is pointed out that directional operators are suitable for the extraction of straight edges [7, 114]. However, rotationally symmetric operators give smooth edges. For instance, as we mentioned earlier, the Laplacian gives smooth, closed edges.

Another important characteristic of edges concerns the discrete nature of the image plane. Deterioration in the performance of an edge detector due to the subpixel (real displacement of the edge from the nearest integer location) is as important as deterioration due to noise [66, 48]. For example, the delocalization error of Canny [8] and Deriche detectors [21] increases as the edge is further away from the integer location (that is, when the subpixel error becomes large). Their signal/noise ratio is also affected by the subpixel, it decreases when the subpixel increases [125]. In general, neglecting the discrete nature of the image plane when evaluating detectors' performance leads to erroneous results and false conclusions.

The influence of edge orientation is related to the rotation invariance property of an edge detector. An edge detector is invariant by rotation iff the differentiation of the smoothing filter is invariant by rotation. As an example, the gradient modulus of the Gaussian is invariant by rotation, whereas the gradient modulus of the 2D version of the filter in (eq. 1) is not. A study of the invariance by rotation property and its influence on the error in edge orientation has been carried out by Davies [16] for several detectors. He proposes a method of implementing a rotationally symmetric detector to reduce the error in edge orientation. By studying several existing detectors, Davies confirms that for the estimation of edge orientation, differential detectors are more accurate than template-matching detectors. Another implementation method was proposed by Merron and Brady [71]. They studied the effect of the gradient of the Gaussian on anisotropy for an ideal step edge. The anisotropy model is identified and used to compute an isotropic gradient estimation. More thorough studies of rotation invariance for gradient and higher-order derivatives have recently been carried out [58, 59, 15]. Lenz [58] shows that rotation-invariant operators are optimal for a wide variety of edge detection schemes and gives a method to design filters for feature extraction. Danielsson and Seger [15] measure the total harmonic distortion generated by spectral analysis of a filter convolved with a test image. Lacroix [55] proposes another criterion based on the variation of the estimated edge orientation, which includes discretization and truncation errors. Rotation invariance also affects the variation of the modulus and the error is the estimated gradient vector's direction. In recent papers [125, 129], we have studied the influence of edge orientation on the gradient vector and the performance (signal-to-noise ratio and delocalization error) of general first-order edge detectors using $(\frac{\partial}{\partial x})^2 + (\frac{\partial}{\partial y})^2$ as a differentiation operator. We start with the influence of the edge orientation $\theta \in (0, \pi/2)$ on the gradient magnitude M and orientation ψ:

$$M(\theta) = \sqrt{((f_x * s_{\theta,l})(x,y)|_{(x,y)=(0,0)})^2 + ((f_y * s_{\theta,l})(x,y)|_{(x,y)=(0,0)})^2} \tag{9}$$

$$\psi = atan\left(\frac{(f_y * s_{\theta,l})(x,y)|_{(x,y)=(0,0)}}{(f_x * s_{\theta,l})(x,y)|_{(x,y)=(0,0)}}\right) \tag{10}$$

where $s_{\theta,l}(x,y)$ is an ideal step edge, l the subpixel and $f(x,y)$ the smoothing filter. We show that the gradient magnitude of rotationally symmetric detectors is unaffected by edge orientation; i.e. $M^{'}(\theta) = 0$. For rotationally dependent detectors, gradient magnitude can be affected by edge orientation; i.e. $M^{'}(\theta) \neq 0$. However, there are rotationally dependent detectors for which gradient magnitude is unaffected by edge orientation, as is the case of rotationally symmetric detectors. Description of the properties of these detectors requires the introduction of some mathematical concepts which are not given here; the reader is referred to paper [129]. The influence of edge orientation on the gradient magnitude is symmetric at an edge orientation equal to $\pi/4$, possibly with an extremum at this orientation. If the extremum exists, its type (minimum or maximum) is defined by the properties of the detector and the characteristics of the edge. When the edge crosses the pixel (subpixel $l = 0$), the estimated edge orientation is accurate for all detectors; i.e., $\theta = \psi$. Otherwise, for rotationally symmetric detectors the estimated edge orientation is also accurate. For rotationally dependent detectors estimation of edge orientation may be biased (i.e., $\theta \neq \psi$), even if the signal is noise-free. However, there are rotationally dependent detectors for which the orientation of the edge is equal to the gradient orientation, as it is for rotationally symmetric detectors.

Now we will deal with the influence of θ on the signal-to-noise ratio and the delocalization error. The signal-to-noise ratio Σ of the first-order edge detector is the quotient of its response to the input signal $c(x, y, \theta)$ and the square root of its mean squared noise response. It is given by:

$$\Sigma^2 = \frac{((f_x^{'} * s_{\theta,l})(x,y)|_{(x,y)=(0,0)})^2 + ((f_y^{'} * s_{\theta,l})(x,y)|_{(x,y)=(0,0)})^2}{n_0^2 \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f_x^{'2}(x,y)dydx + n_0^2 \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f_y^{'2}(x,y)dydx} \tag{11}$$

We assume that the ideal edge passes through the origin $(0, 0)$ and the detected edge is located at (x_0, y_0), where x_0 and y_0 are independent random variables of zero mean. The localization (Λ) can be defined by the following quantity:

$$\Lambda = \frac{1}{E(x_0^2)} + \frac{1}{E(y_0^2)} = \frac{((f_x^{'''} * s_{\theta,l})(x,y)|_{(x,y)=(0,0)})^2}{n_0^2 \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f_x^{''2}(x,y)dydx} + \frac{((f_y^{'''} * s_{\theta,l}(x,y)|_{(x,y)=(0,0)})^2}{n_0^2 \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f_y^{''2}(x,y)dydx} \tag{12}$$

where $E(x^2)$ is the variance of x. The signal-to-noise ratio of rotationally symmetric detectors is unaffected by the orientation of the edge; i.e., $\Sigma^{'}(\theta) = 0$. However, the localization can be affected by edge orientation; i.e., $\Lambda^{'}(\theta) \neq 0$. For rotationally dependent detectors, both signal-to-noise ratio and localization are often affected by edge orientation. However, there are rotationally dependent detectors whose signal-to-noise ratio and/or localization are orientation-free.

These results show that rotationally symmetric detectors do not necessarily provide isotropic information and that rotationally dependent detectors can provide isotropic information. Consequently, they call into question the belief that rotationally symmetric operators compute isotropic information and conversely, isotropic information is computed only by rotationally symmetric operators (see [6]). Furthermore, experimentation has shown that in practice the property of invariance to rotation is not preserved, due to tessellation of the image plane and numerical error approximation. Rotationally dependent detectors are also influenced by these discretization problems and remain highly sensitive to edge orientation. These results do not rule out the use of directionally selective detectors. As we mentioned above, directional detectors are suitable for extracting linear edges whenever they give a better performance for such edges.

4.3 Evaluation of Detectors

Edge detectors provide a set of edge pixels which are combined into more elaborate primitives (*i.e.*, chains, straight lines, circles, splines). The performance of a computer vision or image analysis system depends on this intermediate representation. An edge detector which performs well in a general context produces primitives from which an object may be found with little computation. An edge detector which introduces errors leads to an inefficient system. In spite of major research efforts in this field, edge detectors do not meet the requirements of many applications in computational vision. Detectors miss true edges, detect false edges, and the edge delocalization error is unsatisfactory. These errors depend on image characteristics, detector properties, and implementation methods. Many methods for the analytical study of the influence of other edge characteristics on the performance of detectors have been proposed. The results were summarized in sections 4.1 and 4.2. Here, we will consider experimental evaluation.

Experimental evaluation of the results of an edge detector shows its failures and characterizes its performance. Thus, it makes it possible to distinguish detectors, to refine the mutual influence between characteristics of the detector and those of the image, and can result in a flexible detector (by adjusting its parameters to get useful edges). The evaluation process requires criteria or reference, which describe the characteristics of the edge to be detected. If the criteria are described formally the evaluation is objective; subjective otherwise. Subjective evaluation consists of showing the detected edges to a human subject who rates the detector. While this technique is easy, only a few characteristics (e.g., position, contrast, orientation) are visible to humans. The evaluation is rough since it is difficult for humans to distinguish between two close grey levels or two close orientations. Judgment by humans depends on experience, the context (i.e., the scene) and attachment to the detector used. The human subject does not check whether the detector conforms to its initial specifications but rather whether perceived edges are detected. Subjective evaluations are vague and cannot be used to measure the performance of detectors but only to establish their failure.

The goal of objective evaluation is to measure the performance of an edge detector. Several authors [1, 88, 49, 115, 46] have proposed performance measures to evaluate the output of edge detectors. Abdou and Pratt [1, 88] have proposed a measure, called the figure of merit, which is a combination of three factors: non-detection of true edges, detection of false edges, and edge delocalization error. Using this measure it is difficult to determine the type of error committed by the detector. Kitchen and Rosenfeld's measure [49] combines errors that arise due to an edge's thickness and lack of continuity. Venkatesh and Kitchen [115], among others, use four error types which reflect the major difficulties encountered in edge detection: non-detection of true edges, detection of false edges, detection of several edges instead of an edge one pixel wide, and edge delocalization error. All of these measures have been used empirically to quantify the effect of edge characteristics such as contrast, noise, slope and width on various edge detector schemes. The edge attributes to be considered in the evaluation process are of great interest. The test image should be realistic, especially where knowledge and know-how about realistic camera model are available [51]. It should include different physical phenomena related to edges: multi-models with variable contrast, different models of junctions, blurring, and noise.

Subjective and objective evaluations can be used together to evaluate edge detectors. This combination inspired by psychological methods, is based on statistical analysis. For example, Heath *et al.* [39] propose an evaluation method in the context of object recognition. Edge detector results are presented to humans who compare different edge detectors. They are interpreted using the analysis of variance technique to establish the statistical significance of observed differences. Two experiments are proposed in the paper. The first leads to the automatic computation of parameters of the edge detector. For each edge detector a combination of parameters is chosen and the resulting edges are presented to eight judges. These judges rate the edge detector on a scale of 1 to 7. A rating of 1 means that edges cannot be coherently organized into an object and 7 means that all edges are relevant for recognizing an object. From the first experiment, ratings are analyzed statistically and the best parameters are selected for each edge detector and for each image. The second experiment concerns comparison between edge detectors. Original images and corresponding edges are presented to sixteen judges for rating. The correlation between three factors (edge detector, set of parameters, and image) is analyzed.

To conclude, the results obtained by both experimental and analytical evaluation processes have clarified the mutual influence between edge characteristics and detector properties. More generally, the knowledge that has been acquired about edge detection should be used in the design of edge detectors. Consequently, evaluation methods should be developed with the same interest as the smoothing and differentiation techniques. We suggest that evaluation methods should take into account the subsequent use of edges, the specification of the detector and the characteristics of the real image. Recent results obtained by Heath *et al.* [39] and by Kanungo *et al.* [46] are a promising step in this direction.

5 Survey of Edge Detectors

Since the appearance of image processing, the number of edge detectors has increased continuously. It is difficult to make an inventory of the available algorithms. Most existing edge detectors are autonomous and include the three main steps: smoothing, differentiation and labeling. They differ in their smoothing filters, differentiation operators, labeling processes, goals, computational complexity and the mathematical models used to derive them. Contextual detectors are by far the most rarely used and designed. Their goal is different, as is the knowledge used to extract edges. These detectors are not presented in this paper; however, the reader can find a survey in [17, 68]. Similarly, edge detection approaches based on snakes, statistical tools and neural networks are not presented here. Our goal is not to give an exhaustive inventory of edge detection algorithms. We limit ourselves to edge detection algorithms that fit the detector properties given in previous sections and that have influenced our work over the recent years. Other surveys on edge detection may be found in [17, 114, 123, 75].

In the next sections, we will present informal detectors for step edges or early detectors, optimal step edge detectors, detectors of line edges and junctions, the use of phase information to extract both step edges and line edges, and the implementation of edge detectors.

5.1 Detection of Step Edges

As much as thirty years ago, the first detectors based on gradient and Laplacian operators were proposed. These detectors are limited to the differentiation operation. For instance, estimation of the gradient vector is based on the use of the following 3×3 masks:

$$\Delta_x = \begin{bmatrix} -1 & 0 & 1 \\ -a & 0 & a \\ -1 & 0 & 1 \end{bmatrix} \qquad \text{and} \qquad \Delta_y = \begin{bmatrix} -1 & -a & -1 \\ 0 & 0 & 0 \\ 1 & a & 1 \end{bmatrix}$$

where a is a positive real number (1 in the case of Prewitt's masks [89] and 2 in the case of Sobel's masks [66]). The Prewitt and Sobel masks are well-known operators in edge detection. The intensity image is convolved with each mask to compute the first-order partial derivatives at the center of a 3×3 window. For the Laplacian estimation, the following 3×3 mask is used:

$$\nabla = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

The masks most often proposed have a fixed 3×3 size. The performance of these operators deteriorates when the image is noisy since, for example, both Prewitt's and Sobel's masks are derived by assuming that white noise is additive and image surfaces are linear. Rosenfeld and Thurston [94] introduce the smoothing operation to reduce the noise of the image and thus improve image differentiation. Smoothing is done by replacing the value of the pixel by the average computed on a squared window. The drawback of the differentiation operators mentioned above is that their size and coefficients are fixed, so they cannot be adapted to a given image. Therefore, they remain noise-sensitive. Furthermore, these detectors are informal, in the sense that no formal model is used to represent edges or to derive them. Thus, analysis of their performance is based on informal criteria.

The problems underlying the use of the smoothing operation concern the choice of the appropriate filter and its scale. The problem of scale selection has engendered multi-scale edge detection (Cf. 3.4). It should be recalled that smoothing has a positive effect, noise reduction, thus ensuring robust edge detection, and a negative effect, loss of information. Clearly, we have a fundamental trade-off here between information loss and noise reduction. The ultimate goal is to find optimal detectors that ensure an acceptable compromise between noise reduction and edge conservation. Each of the optimal detectors proposed in the literature falls into one of two categories: parametric fitting and optimal enhancement.

Parametric fitting: these techniques involve adjusting the image by an edge model and selecting the edge pixels for which the fitting error is minimal. The differentiation problem is thus avoided. To illustrate, we will begin by describing Hueckel's technique [42, 43]. Each image pixel is fitted by a two-dimensional step edge in a circular window. The parameters of this edge model are the luminance level, the edge orientation and the distance from the center. If the fit is sufficiently accurate, an edge is assumed to exist with the same parameters as the ideal edge model. The accuracy of edge fitting is measured in terms of the mean square error criterion. Hueckel introduces a polar Fourier expansion and uses the first eight coefficients in the minimization procedure. Although this approximation simplifies the computation model needed, it affects the accuracy of the minimization procedure. Other researchers have been interested in parametric fitting techniques for edge detection [43, 106, 23, 74, 76, 77]. Since these techniques are a rich description of the image structures, they have the advantage of providing all edge attributes such as subpixel position, contrast, blur, width, and intensity level. For instance, in the recent work of Nayar *et al.* [77], the image intensity is correlated with a given edge model and all its attributes are computed. To avoid inefficient computation, the number of edge attributes is reduced. For example, a 2D step edge, $u(x, y) = c_1$ if $xcos(\theta)+ysin(\theta)+\rho \geq 0$ and c_2 elsewhere, has four parameters, namely, orientation θ, subpixel ρ, and two brightness values c_1 and c_2. The normalized step edge $\frac{(u(x,y)-c_2)}{c_1-c_2}$ is independent of c_1 and c_2. The authors go further in dimension reduction with little loss of information. Instead of the normalized edge, they use a dimensionless edge parametric manifold obtained from the significant eigenvectors of the initial edge model. The edge parametric manifold is compared to image pixels. If the distance between the manifold and an image window is sufficiently small, there is an edge. The parameters of the manifold are used to estimate the parameters of the initial edge model. This scheme is used for the extraction of five edge models: step, line, roof, corner, and blob.

Another technique, different from the preceding has been proposed by Haralick [36, 37]. It consists in adjusting the image by a set of given basis functions. The differentiation problem is easy to resolve since we are computing derivatives of continuous functions. Step edges occur at pixels having a negatively sloped zero-crossing of the second directional derivative taken in the direction of the gradient. The image is fitted by a linear combination of discrete bases of Tchebychev's polynomial, of order less than or equal to three. First and second-order derivatives are computed and used to locate edges. It is pointed out in [7] that fitting by this polynomial bases of order less than or equal to three, is equivalent to the use of smoothing with the Gaussian filter (which is optimal according to regularization requirements).

Optimal enhancement: this involves designing edge detectors with desired performance. There are two problems related to optimal edge detection: the definition of performance criteria and the design of a filter which optimizes these criteria. Usually, the definition of performance depends on detection accurately and delocalization error of edges. The design of optimal detectors requires the specification of these criteria to give mathematical models, and the use of optimization theory to derive the optimal detector according to these criteria. Specification of these criteria by different authors gives different mathematical models and thus different optimal detectors.

Shanmugam *et al.* [98] consider the problem of optimizing spatial frequency domain filters within a finite interval in the vicinity of edge pixels. Using this criterion, authors derive a band limited filter which has a form similar to the Laplacian of Gaussian. Surprisingly, edges are located at the extrema of this detector output. Consequently, this detector responds by two extrema to a single edge pixel. For example, when the 1D smoothed step edge is the $erf(x)$ function, the absolute value of its second derivative (i.e, $\frac{4}{\sqrt{\pi}}|x|exp(-x^2)$) has two maxima. In contrast to Shanmugam *et al.*, Marr and Hildreth [67] and later Hildreth [41] have proposed the use of zero-crossings of the Laplacian of Gaussian. The image is convolved with the Laplacian of the two-dimensional Gaussian and the zero-crossings are labeled. The choice of the Gaussian filter is motivated by the fact that it represents an optimal compromise between spatial and frequential resolution. Moreover, the Laplacian of Gaussian can be implemented efficiently, for example, as the difference of two Gaussians having two close scales, or by computing only two partial derivatives. Figure 5 presents step edges obtained by the Laplacian of Gaussian.

Another optimal edge detector is proposed by Canny [7, 8]. He assumes that edge detection is to be performed by convolving the image with a filter, and marking edges at the output maxima. He derives an optimal filter for the extraction of a one-dimensional ideal step edge in the presence of white Gaussian noise. The performance criteria are good detection, good localization, and an unique response to a single edge. According to Canny, this filter can be accurately approximated by the first derivative of the Gaussian. To create a two-dimensional edge detection scheme, the image is convolved with the first-order derivatives of the two-dimensional

Gaussian. Thus, the gradient of the smoothed image is computed and edges are located at the maxima of the gradient modulus taken in the direction of the gradient. Canny's work has inspired many researchers [100, 21, 83, 96, 57, 105]. Deriche [21] extended Canny's initial filter to two dimensions and implemented it using recursive filtering (cf. 5.5). By using an ideal step edge and similar performance criteria to those of Canny, Shen and Castan [100, 101] derived an exponential filter and implemented it using recursive filtering. In [9, 125, 129] the reader will find a theoretical evaluation of Deriche's, Canny's, and Shen and Castan's detectors. Sarkar and Boyer [96, 97, 113] propose an optimal infinite-response edge detection filter using an ideal step edge and Canny's criteria. Petrou and Kittler [83] derive another optimal detector using criteria similar to those of Canny but a blurred step edge model.

All these detectors based on optimal enhancement provide integer location of edges. Additional schemes for the computation of real location of edges can be found in [44, 110, 108, 47]. Properties of detectors that reduce the negative effect of subpixel are given in [129]. In addition, these detectors consider that a step edge is a local discontinuity of the grey level function. There are many problems with this definition. A visual examination of the image shows that subjective contours do not correspond to local discontinuities [117]. Another problem is that the edge is defined as a single point, without consideration of neighboring edges. Moreover, most previously reported efforts have proposed 1D models of edges and consequently factors such that the rotation by invariance property are not considered. A possible improvement in the definition of edge is the use of a 2D multi-model (i.e., multi-step) to include other phenomena (e.g., the Laplacian of Gaussian gives a false zero-crossing in response to a staircase edge model). Recent progress has been achieved in this respect; 1D multi-model edge detectors are proposed in [10, 81, 99] and a 2D optimal step edge Laplacian detector is proposed in [70]. Finally, as we mentioned earlier the step edge is by far the most common. However, there are other local variations of the grey level function (e.g., roof, junction, shoulder) which have a meaning in our world and which can be extracted and used in computer vision systems. Taking these variations into account leads to a rich description of the world and therefore an improvement in computer vision. However, as we will explain below (cf. 5.4 and 3.4) the integration of these different variations is a new problem.

5.2 Detection of Lines

As mentioned earlier, lines correspond to local extrema of the grey level image and are of great use in the identification of image features, such as roads and rivers in remote sensing images for example. Most schemes for the detection of lines are limited to thinning algorithms. The majority of these algorithms are designed for binary images [2, 11, 103] and a few for grey level images [26, 95]. The main problem is that they usually yield edges which are not located accurately enough and they do not perform well in complex images such as remote sensing images.

Haralick [36] proposed an algorithm based on polynomial fitting. The basic idea of this algorithm is similar to the author's step edge detection algorithm described earlier. The image is fitted by a linear combination of discrete bases of Tchebychev's polynomial of order less than or equal to three. Lines occur at pixels having zero-crossings of the first directional derivative taken in the direction that maximizes the second directional derivative.

Giraudon [33] proposed an algorithm for detecting a line at a negative local maximum of the second derivative of the image, rather than a zero-crossing of the first derivative as in the Haralick case. He estimated the second derivative by convolving the image with the difference of two Gaussians having close scales. The search for a negative maximum is performed along the gradient direction. The main problem with Giraudon's detector comes from the use of the gradient since at the peak point, the gradient value is too small to be used.

Using a 1D ideal roof model and Canny's criteria, Ziou [124] derives an optimal line detector. In 2D, the image is convolved with two directional filters operating in the x direction and in the y direction separately. The resulting images are combined and lines are located in this image at the maxima in the direction that maximize the grey level variance. As in the case of step edge detection [20, 100, 96], this line detector is efficiently implemented using recursive filtering. Figure 9 presents line edges obtained by this detector.

Koundinya and Chanda [53] have proposed an algorithm-based combinatorial search. The basic idea behind this algorithm is to locate lines that maximize an *ad hoc* confidence measure. The confidence measure of a candidate pixel is proportional to the number of pixels in its vicinity having a different grey level than the candidate pixel. Authors have experimented the three strategies for combinatorial search: conventional tracking, best-first and depth-first. According to the results provided in the paper, the best-first strategy seems to provide a more complete edge.

a. b.

Figure 9: *a) Original Image, b) Line edges.*

5.3 Detection of Junctions

Junctions are extremely useful features. They are very helpful in solving correspondence problems in computer vision. Interest in the junction extraction is growing [78, 79, 61, 35, 69, 28, 91, 14, 92]. Rangarajan *et al.* [90] define a junction as the intersection of two linear step edges. Using Canny's criteria, they derive an optimal detector. This detector locates only one junction model (junctions formed by two linear edges symmetric relative to the x axis). To extract junctions in real images, they consider twelve configurations which differ in the orientation of the two linear edges and the angle between them. Consequently, the image is convolved with twelve different masks and junctions are located at the local maxima of the convolution output.

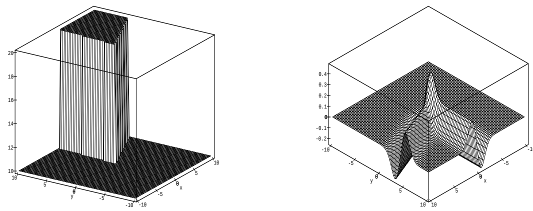
In the tradition of Hueckel, one approach recently proposed by Rohr [92] involves fitting a general blurred junction model to the observed image. The author starts with the identification of the edge model (X, T, L, and so on) by computing the number of adjacent regions in a working window. If the fit is sufficiently accurate, a junction edge is assumed to exist with the same parameters as the blurred junction model. The author has focused much effort on studying the behavior of the blurred junction model, the minimization method for least square fitting, and on selection of the working window since the performance of the algorithm is dependent on this window.

Deriche and Giraudon [24] have proposed the extraction of junctions at zero-crossings of the Laplacian of Gaussian of the image. The authors show that a local maximum of the Hessian determinant of the smoothed image moves in scale space along a line that passes through the exact position of the central point of the vertex. This property is used by authors to choose junctions between all zero-crossings of the Laplacian; they retain zero-crossings that occur on these lines. Instead of using the Hessian, Tabbone [107] shows that the Laplacian of Gaussian of the image presents an elliptic extremum which always lies inside the corner as shown in Fig. 10. This extremum moves in scale space along the line that bisects the corner. As above, the junction is a zero-crossing of the Laplacian that occurs on the line that passes through the extremum located at two different scales. Figure 11 presents the junctions detected by this detector.

Recently, Kohlmann [50] has proposed an efficient corner detector using the Hilbert transform. L, T, and Y junctions are located as maxima of the Hilbert transform of the image. The main effort of the author concerns the efficient implementation of the Hilbert transform by a separable 2D filter.

5.4 Local Energy and Phase Congruency

Since an image contains many types of edges, consideration of these edge models allows us to iron out many problems in computer vision and image processing. To extract these edges,



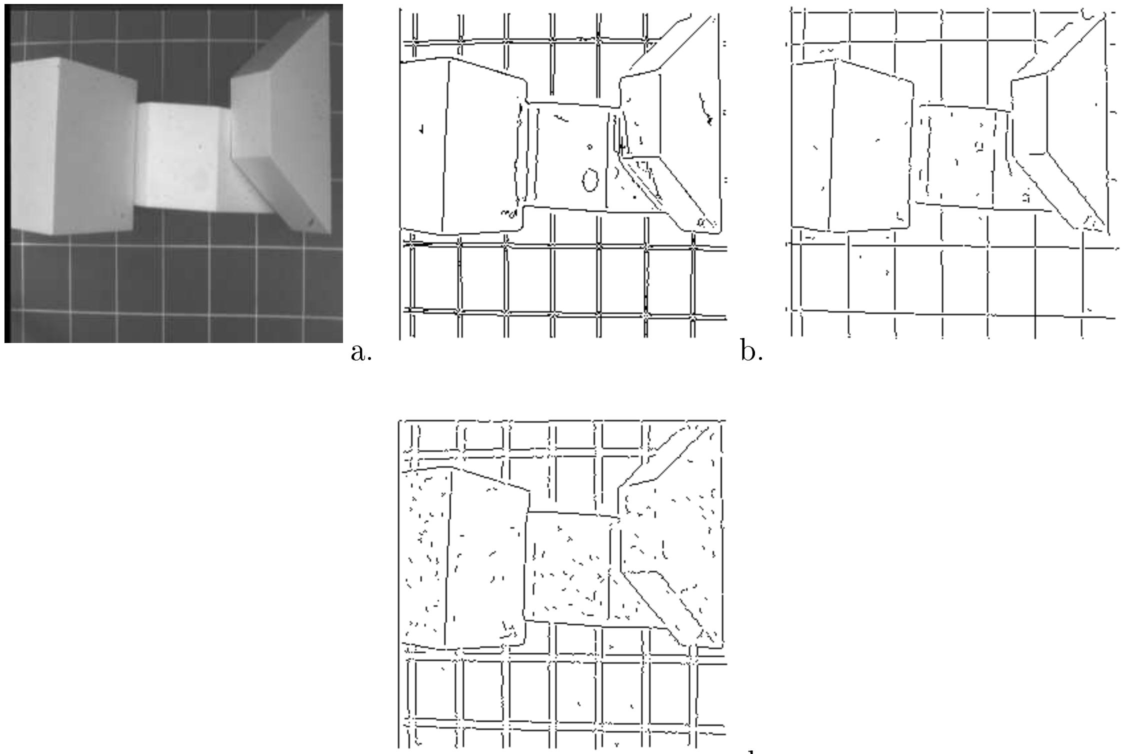
a. b.

Figure 10: *a) L-Junction, b) The output of the Laplacian of Gaussian.*

Figure 11: *Junctions extracted at the zero-crossings of the Laplacian of Gaussian.*

one could for example run, separately, a step detector and line detector, and combine their outputs. This process is inadequate since it leads to edge duplication [80, 124, 93]. In fact, a step edge detector locates two neighboring steps in a line as shown in Fig. 12.b. Usually, a line edge detector localizes two neighboring minima (resp. maxima) in a maximum (resp. minimum) of the grey level and also locates one or two neighboring extrema in a step as shown in Figs. 12.c and 12.d.

To avoid matching of edges some authors [80, 54, 81, 32, 116, 73] propose to combine the outputs of a symmetric filter (i.e., second derivative of the Gaussian) and an anti-symmetric filter (i.e., first derivative of the Gaussian). Gaussian filters are not the only ones which can be used in these edge detection schemes; quadratic filters can also be used. Let G and H be the Fourier transforms of g and h. Then, g and h are in quadrature if they are Hilbert transforms of each other; namely $H(w) = -jG(w)sign(w)$, where $sign(w)$ is 1 for $w > 0$, 0 for $w = 0$ and -1 otherwise. To illustrate, the real and imaginary part of the Fourier transform of a causal function form a Hilbert transform pair. It is not easy to justify the use of quadrature pairs of filters in edge detection. Perhaps their use simplifies mathematical handling since they are orthogonal. Recently, Ronse [93] has provided additional requirements that must be fulfilled by the quadrature pair filters to avoid detecting false maxima in scale space and to reduce the delocalization error.



a. b. c. d.

Figure 12: *a) Original Image, b) Step edges, c) Line edges, d) Line edges on the negative of the image.*

The output of a quadrature pair of filters is combined to provide energy and phase information. The energy, or amplitude, is the sum of squares of convolutions of the image with a symmetric filter and an anti-symmetric filter. The phase (the arctan of the anti-symmetric filter output divided the symmetric filter output) is also used in many proposed schemes for the extraction and classification of edges [73, 116]. For instance, at the origin the phase of an ideal step edge smoothed by the Gaussian is $\pi/2$, whereas the phase of a delta function smoothed by the Gaussian is 0. Perona and Malik [81] locate edges at the energy maxima. In their schemes, they use odd and even symmetric directional filters at different orientations, and the orientation that maximizes the quadratic output is retained as the edge orientation. Using a steering theorem, Freeman [32] presents an efficient implementation method for an oriented filter in any direction as a linear combination of oriented filters in particular directions. This implementation method has been used in the detection of lines [25]. Figure 13 presents the output of step edge and the output of edge detector based on quadrature pair of filters.

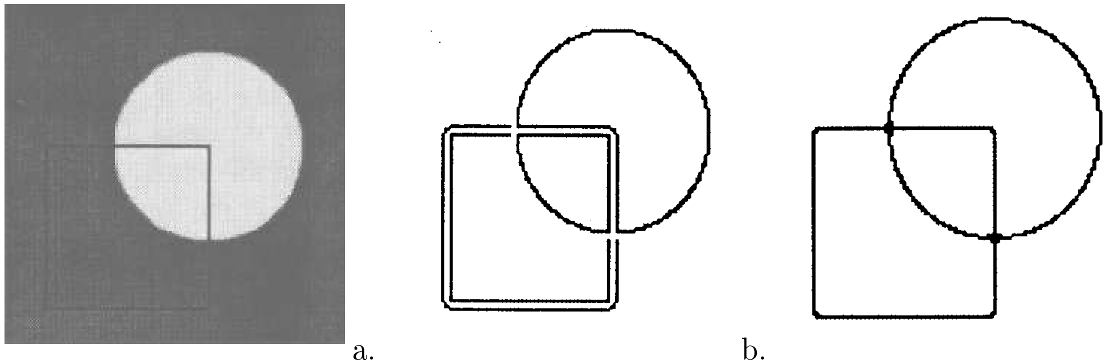


Figure 13: *a) Original Image, b) The output of step edge detector, c) The output of an edge detector based on quadrature pair of filters.*

5.5 Implementation of Detectors

The implementation of an edge detector is a non-trivial problem. Usually, the design of an edge detector is the result of a theoretical analysis and many refinements are required to get a running program. Many implementation methods have been proposed [102, 31, 104] such as convolution masks, Fourier transform, hierarchical correlation, and numerical filtering among others. The fundamental question concerns the desirable requirement of an implementation method for a given detector. It is obvious that efficiency is an important criterion since the edge detector must run in a reasonable amount of time. In the case of multi-scale edge detection, this requirement also means that increasing the scale must not drastically affect the computation time. For instance, the convolution mask method is sensitive to scale increases. In addition, the implemented detector must preserve the properties of the detector resulting from theoretical analysis. In fact, when edge detection involves a convolution operation of the image and a continuous filter the implementation requires sampling of this filter into its discrete form. Moreover, a cut-off of an infinite support filter is required when it is implemented using methods such as convolution masks and Fourier transform. These approximations yield aliasing phenomena and result in loss of information. One can reduce the effect of these phenomena and thus preserve the detector properties by a judicious choice of appropriate procedures for sampling, quantization and the cut-off of the filter. These aspects have been considered by investigators [104, 31, 44, 102] for the implementation of the Laplacian of Gaussian. Another information loss problem arises from image border erosion. When convolving an image of size $N \times N$ with a mask of size $M \times M$, we normally lose $2MN - M^2$ pixels from the border. For example, when the Laplacian of Gaussian is employed with $\sigma = 3$, M is set at $8 * \sqrt{2}\sigma \simeq 33$ and $N = 256$, causing a 24% loss in image area. The method using an infinite impulse response filter (IIR) has been used to implement step edge detectors [72, 21, 22, 96, 97] and a line edge detector [124]. This method has multiple advantages. No filter cut-off is necessary. The computational effort required to smooth images and compute derivatives is drastically reduced. The computational measures are unaffected by the scale of the detector. Therefore, the use of recursive filtering is more suitable for multi-scale edge detection. However, recursive filtering (IIR) is strictly sequential (that is, the output at one pixel depends on the output at some neighboring pixels) and it is not suitable when we are interested in detecting only edges at specific location in the image rather than in the whole image. A comparison of convolution masks and recursive filtering for the detection of lines is given in [124].

References

[1] I.E Abdou. Quantitative Methods of Edge Detection. Technical report no. 830, Image Processing Institute, University of Southern California, 1978.

[2] C. Arcelli and G. Saniti di Baja. A Width-Independent Fast Thinning Algorithm. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 7(4):463–474, July 1985.

[3] J. Babaud, A.P Witkin, M. Baudin, and R.O Duda. Uniqueness of the Gaussian Kernel for Scale-Space Filtering. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 8(1):26–33, Jan 1986.

[4] F. Bergholm. Edge Focusing. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 9(6):726–741, Nov 1987.

[5] V. Berzins. Accuracy of Laplacian Edge Detectors. *Computer Vision, Graphics and Image Processing*, 27:195–210, 1984.

[6] M. Brady and B.K.P. Horn. Rotationally Symmetric Operators for Surface Interpolation. *Computer Vision, Graphics and Image Processing*, 22:70–94, 1983.

[7] J.F. Canny. Finding Edges and Lines in Images. Technical report no. 720, Massachusetts Institute of Technology, 1983.

[8] J.F. Canny. A Computational Approach to Edge Detection. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, Nov 1986.

[9] S. Castan, J. Zhao, and J. Shen. Optimal Filter for Edge Detection Methods and Results. In *Proceedings of the First European Conference on Computer Vision (*ECCV*)*, pages 13–17, 1990.

[10] J.S. Chen and G. Medioni. Detection, Localisation, and Estimation of Edges. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 11(2):191–198, 1989.

[11] R.T. Chin, H.K. Wan, D.L Stover, and R.D Iverson. A One-Pass Thinning Algorithm and its Parallel Implementation. *Computer Vision, Graphics and Image Processing*, 40:30–40, 1987.

[12] J.J. Clark. Singularity Theory and Phantom Edges in Scale Space. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 10(5):720–727, Sept 1988.

[13] J.J. Clark. Authenticating Edges Produced by Zero-Crossing Algorithms. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 11(1):43–57, June 1989.

[14] J. Cooper, S. Venkatesh, and L. Kitchen. Early Jump-Out Corner Detectors. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 15(8):823–828, 1993.

[15] P.E. Danielsson and O. Seger. Rotation Invariance in Gradient and Higher Order Derivative Detectors. *Computer Vision, Graphics and Image Processing*, 49:198–221, 1990.

[16] E.R. Davies. Circularity-A New Principle Underlying the Design of Accurate Edge Orientation Operators. *Image and Vision Computing*, 2(3):134–142, 1984.

[17] L.S. Davis. A Survey of Edge Detection Techniques. *Computer Graphics and Image Processing*, 4:248–270, 1975.

[18] S. Dellepiane, D.D. Guisto, and G. Vernazza. Automatic Parameter Computation for Edge Detection by the Zero-Crossing Method. In 12^{th} *Conference on Signal Processing and Images (*GRETSI), pages 617–620, France, 1989.

[19] E. DeMicheli, B. Caprile, P. Ottonello, and V. Torre. Localization and Noise in Edge Detection. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1107–1116, 1989.

[20] R. Deriche. Optimal Edge Detection Using Recursive Filtering. In *Proceedings of the First International Conference on Computer Vision*, pages 501–505, Londres, 1987.

[21] R. Deriche. Using Canny's Criteria to Derive a Recursive Implemented Optimal Edge Detector. *The International Journal of Computer Vision*, 1(2):167–187, 1987.

[22] R. Deriche. Fast Algorithm for Low-Level Vision. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 12(1):78–87, 1990.

[23] R. Deriche and T. Blaszka. Recovering and Characterizing Image Features Using an Efficient Model Based Approach. In *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*, USA, 1993.

[24] R. Deriche and G. Giraudon. A Computational Approach for Corner and Vertex Detection. *The International Journal of Computer Vision*, 10(2):101–124, 1993.

[25] J.C. DiMartino and S. Tabbone. An Approach to Detect Lofar Lines. *Pattern Recognition Letters*, 17:37–46, 1996.

[26] C.R. Dyer and A. Rosenfeld. Thinning Algorithms for Gray-Scale Pictures. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 1(1):88–89, Jan 1979.

[27] J.H. Elder and S.W. Zucker. Scale Space Localization, Blur, and Contour-Based Image Coding. In *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*, pages 27–34, San Francisco, 1996.

[28] R. Espelid and I. Jonassen. A Comparison of Splitting Methods for the Identification of Corner-Points. *Pattern Recognition Letters*, 12(2):79–83, 1991.

[29] M.A. Fischler, J.M. Tenenbaum, and H.C. Wolf. Detection of Roads and Linear Structures in Low-Resolution Aerial Imagery Using a Multisource Knowledge Integration Technique. *Computer Vision, Graphics and Image Processing*, 15:201–223, 1981.

[30] M. M. Fleck. Multiple Widths Yield Reliable Finite Differences. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 14:412–429, 1992.

[31] M.R.B. Forshaw. Speeding Up the Marr-Hildreth Edge Operator. *Computer Vision, Graphics and Image Processing*, 41:172–185, 1988.

[32] W.T. Freeman and E.H. Adelson. The Design and Use of Steerable Filters. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[33] G. Giraudon. Edge Detection from Local Negative Maximum of Second Derivative. In *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*, pages 643–645, San-Francisco Ca. (USA), 1985.

[34] M. Gokmen and C.C. Li. Edge Detection and Surface Reconstruction Using Refined Regularization. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 15(5):492–499, 1993.

[35] A. Guiducci. Corner Characterization by Differential Geometry Techniques. *Pattern Recognition Letters*, 8(5):311–318, 1988.

[36] R.M. Haralick. Ridge and Valley on Digital Images. *Computer Vision, Graphics and Image Processing*, 22:28–38, 1983.

[37] R.M. Haralick. Digital Step Edges from Zero Crossing of Second Directional Derivatives. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 6(1):58–68, Jan 1984.

[38] J. Hasegawa, H. Kubota, and J. Toriwaki. Automated Construction of Image Processing Procedures by Sample-Figure Representation. In *Proceedings of the* 8^{th} *International Conference on Pattern Recognition*, pages 586–588, Paris, 1986.

[39] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer. Comparison of Edge Detectors: A Methodology and Initial Study. In *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*, pages 143–148, San Francisco, 1996.

[40] A. Herskovitz and T. Binford. On Boundary Detection. M.I.T, Al Memo 183, 1970.

[41] E.C. Hildreth. The Detection of Intensity Changes by Computer and Biological Vision Systems. *Computer Vision, Graphics and Image Processing*, 22:1–27, 1983.

[42] M.H. Hueckel. An Operator Which Locates Edges in Digitized Pictures. *J. Assoc. Comput. Mach.*, 19(1):113–125, 1971.

[43] M.H. Hueckel. A Local Visual Operator Which recognizes Edges and Lines. *J. Assoc. Comput. Mach.*, 20(4):634–647, 1973.

[44] A. Huertas and G. Medioni. Detection of Intensity Changes with Subpixel Accuracy Using Laplacian-Gaussian Masks. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 8(5):651–664, Sep 1986.

[45] H. Jeong and C.I Kim. Adaptive Determination of Filter Scales for Edge Detection. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, pages 579–585, May 1992.

[46] T. Kanungo, M.Y. Jaisimba, J. Palmer, and R.M. Haralick. A Methodology for Quantitative Performance Evaluation of Detection Algorithms. IEEE *Transactions on Image Processing*, 12(4):1667–1674, 1995.

[47] M. Kisworo, S. Venkatesh, and G. West. Modeling Edges at Subpixel Accuracy Using the Local Energy Approach. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 16(4):405–410, 1994.

[48] L. Kitchen and J.A. Malin. The Effect of Spatial Discretization on the Magnitude and Direction Response of Simple Differential Edge Operators on a Step Edge. *Computer Vision, Graphics and Image Processing*, 47:243–258, 1989.

[49] L. Kitchen and A. Rosenfeld. Edge Evaluation Using Local Edge Coherence. IEEE *Transactions Systems, man, and cybernetics*, SMC-11(9):597–605, 1981.

[50] K. Kohlmann. Corner Detection in Natural Images Based on the 2-D Hilbert Transform. *Signal Processing*, 48:225–234, 1996.

[51] C. Kolb. A Realistic Camera Model for Computer Graphics. In *ACM/SIGGRAPH Computer Graphics Proceedings, Annual Conference Series*, pages 317–324, 1995.

[52] A.F. Korn. Toward a Symbolic Representation of Intensity Changes in Images. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 10(5):610–625, 1988.

[53] K.N.V.L.N. Koundinya and B. Chanda. Detecting Lines in Gray Level Images Using Search Techniques. *Signal Processing*, 37:287–299, 1994.

[54] P. Kube. Properties of Energy Edge Detectors. In *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*, 1992.

[55] V. Lacroix. Edge Detection: What About Rotation Invariance? *Pattern Recognition Letters*, 11:797–802, Dec 1990.

[56] V. Lacroix. The Primary Raster: A Multiresolution Image Description. In *Proceedings of the* 10^{th} *International Conference on Pattern Recognition*, pages 903–907, 1990.

[57] D. Lee, G.W. Wasilkowski, and R. Mehrotra. A New Zero-Crossing-Based Discontinuity Detector. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 2:265–268, 1993.

[58] R. Lenz. Optimal Filters for the Detection of Linear Patterns in 2-D and Higher Dimensional Images. *Pattern Recognition*, 20:163–172, 1987.

[59] R. Lenz. Group-Theoretical Model of Feature Extraction. *J. Opt. Soc. Am. A*, 6:827–834, 1989.

[60] T. Lindeberg. Discrete Derivative Approximations with Scale-Space Properties: A Basis for Low-Level Feature Extraction. *Journal of Mathematical Imaging and Vision*, 3:349–376, 1993.

[61] T. Lindeberg. Junction Detection with Automatic Selection of Detection Scales and Localization Scales. In *Proceedings of the IEEE International Conference on Image Processing*, pages 924–928, Austin, USA, 1994.

[62] T. Lindeberg. Edge Detection and Ridge Detection with Automatic Scale Selection. In *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*, pages 465–470, San Francisco, 1996.

[63] Y. Lu and R.C. Jain. Behavior of Edges in Scale Space. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 11(4):337–356, Apr 1989.

[64] Y. Lu and R.C. Jain. Reasoning about Edges in Scale Space. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 14(4):450–468, 1992.

[65] W.H.H.J. Lunscher and M.P Beddoes. Optimal Edge Detector Design I : Parameter Selection and Noise Effects. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 8(2):164–177, Mar 1986.

[66] E.P. Lyvers and O.R. Mitchell. Precision Edge Contrast and Orientation Estimation. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 10(6):927–937, 1988.

[67] D. Marr and E.C. Hildreth. Theory of Edge Detection. *Proceedings of the Royal Society of London B207*, pages 187–217, 1980.

[68] T. Matsuyama. Expert System for Image Processing: Knowledge-Based Composition of Image Analysis Processes. *Computer Vision, Graphics and Image Processing*, 48:22–49, 1989.

[69] R. Mehrotra, S. Nichani, and N. Ranganathan. Corner Detection. *Pattern Recognition*, 23:1223–1233, 1990.

[70] R. Mehrotra and S. Zhan. A Computational Approach to Zero-Crossing-Based Two Dimensional Edge Detection. *CVGIP: Graphical Models and Image Processing*, 58:1–17, 1996.

[71] J. Merron and M. Brady. Isotropic Gradient Estimation. In *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*, pages 652–659, San Francisco, 1996.

[72] J.W. Modestino and R.W Fries. Edge Detection in Noisy Images Using Recursive Digital Feltering. *Computer Graphics and Image Processing*, 6:409–433, 1977.

[73] M.C. Morrone and R.A. Owens. Feature Detection from Local Energy. *Pattern Recognition*, 6:303–313, 1987.

[74] V.S. Nalwa. Edge-Detector Resolution Improvement by Image Interpolation. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 9(3):446–451, 1987.

[75] V.S. Nalwa. *A Guided Tour of Computer Vision*. Addison-Wesly Publishing Company, 1993.

[76] V.S. Nalwa and T.O. Binford. On Detecting Edges. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 8(6):699–713, Nov 1986.

[77] S.K. Nayar, S. Baker, and H. Murase. Parametric Feature Detection. In *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*, pages 471–477, San Francisco, 1996.

[78] J.A. Noble. Finding Corners. *Image and Vision Computing*, 6(4):121–128, 1988.

[79] J.A. Noble. Finding Half Boundaries and Junctions in Images. *Image and Vision Computing*, 10(4):219–232, 1992.

[80] R. Owens, S. Vankatesh, and J. Ross. Edge Detection is a Projection. *Pattern Recognition Letters*, 9:233–244, 1989.

[81] P. Perona and J. Malik. Detecting and Localizing Edges Composed of Steps, Peaks and Roofs. In *Proceedings of the Third International Conference on Computer Vision*, Japan, 1990.

[82] P. Perona and J. Malik. Scale-Space and Edge Detection Using Anisotropic Diffusion. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[83] M. Petrou and J. Kittler. Optimal Edge Detector for Ramp Edges. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 13(5):483–491, 1991.

[84] I. Pitas and A.N. Venetsanopoulos. Edge Detector Based on Nonlinear Filters. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 8(4):538–550, 1986.

[85] T. Poggio and V. Torre. Ill-Posed Problems and Regularization Analysis in Early Vision. Artificial Intelligence Lab. Memo, No. 773, Massachusetts Institut of Technology, 1984.

[86] T. Poggio and V. Torre. A Regularized Solution to Edge Detection. Artificial Intelligence Lab. Memo, No. 833, Massachusetts Institut of Technology, 1985.

[87] J. Ponce and M. Brady. Toward a Surface Primal Sketch. In *Three Dimensional Vision*, T. Kanade, Ed. New York: Academic Press, 1985.

[88] W.K. Pratt. *Digital Image Processing.* Wiley-Interscience Publication, 1978.

[89] J.M.S Prewitt. Object Enhancement and Extraction. In *In Picture Processing and Psychopictorics*, B.S. Lipkin and al., Eds. Academic Press, New York, 1970.

[90] K. Rangarajon, M. Shah, and D.V. Brackle. Optimal Corner Detector. *Computer Vision, Graphics and Image Processing*, 48:230–245, 1989.

[91] A. Rattarangsi and R.T. Chin. Scale-Based Description of Corners of Planar Curves. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 14(4):430–449, 1992.

[92] K. Rohr. Recognizing Corners by Fitting Parametric Models. *The International Journal of Computer Vision*, 9(3):213–230, 1992.

[93] C. Ronse. The Phase Conguence Model for Edge Detection in Two-Dimensional Pictures: a Mathematical Study. Technical Report, Rap95-11, Universite de Strasbourg, LSIIT - URA 1871, 1995.

[94] A. Rosenfeld and M. Thurston. Edge and Curve Detection for Visual Scene Analysis. IEEE *Transactions on Computer*, 20(5):562–569, May 1971.

[95] E. Salari and P. Siy. The Ridge-Seeking Method for Obtaining the Skeleton of Digital Images. IEEE *Transactions Systems, man, and cybernetics*, 14(3):524–528, 1984.

[96] S. Sarkar and K.L. Boyer. On Optimal Infinite Impulse Response Edge Detection Filters. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1154–1171, 1991.

[97] S. Sarkar and K.L. Boyer. Comments on "On Optimal Infinite Impulse Response Edge Detection Filters". IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 16(1):106–108, 1994.

[98] K.S. Shanmugam, F.M. Dickey, and J.A. Green. An Optimal Frequency Domain Filter for Edge Detection in Digital Pictures. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 1(1):37–49, Jan 1979.

[99] J. Shen. On Multi-Edge Detection. *CVGIP: Graphical Models and Image Processing*, 58(2):101–114, Mar 1996.

[100] J. Shen and S. Castan. An Optimal Linear Operator for Edge Detection. In *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*, pages 109–114, Miami, 1986.

[101] J. Shen and S. Castan. An Optimal Linear Operator for Step Edge Detection. *CVGIP: Graphical Models and Image Processing*, 54(2):122–133, Mar 1992.

[102] J. Shen and W. Shen. Image Smoothing and Edge Detection by Hermite Integration. *Pattern Recognition*, 28(8):1159–1166, 1995.

[103] R.W. Smith. Computer Processing of Line Images: A Survey. *Pattern Recognition*, 20(1):7–15, 1987.

[104] G.E. Sotak and K.L. Boyer. The Laplacian-of-Gaussian Kernel: A Formal Analysis and Design Procedure for Fast, Accurate Convolution and Full-Frame Output. *Computer Vision, Graphics and Image Processing*, 48:147–189, 1989.

[105] L.A. Spacek. Edge Detection and Motion Detection. *Image and Vision Computing*, 4:43–56, 1986.

[106] A.J. Tabatai and O.R. Mitchell. Edge Location to Subpixel Values in Digital Imagery. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 6(2):188–201, 1984.

[107] S. Tabbone. Detecting Junctions Using Properties of Gaussian Detector. In *Proceedings of the* 12^{th} *International Conference on Pattern Recognition*, pages 52–56, Israel, 1994.

[108] S. Tabbone. Edge Detection, Subpixel and Junctions Using Multiple Scales. PhD thesis, Institut National Polytechnique de Lorraine, France, *(In french)*, 1994.

[109] S. Tabbone and D. Ziou. Elimination of False Edges by Separation and Propagation of Thresholds. In 13^{th} *Conference on Signal Processing and Images (*GRETSI), pages 1025–1028, *France*, 1991.

[110] S. Tabbone and D. Ziou. Subpixel Positioning of Edges for First and Second Order Operators. In *Proceedings of the* 11^{th} *International Conference on Pattern Recognition*, pages 655–658, Netherlands, 1992.

[111] S. Tabbone and D. Ziou. Efficient Edge Detection Using Two Scales. In *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*, 1993.

[112] S. Tabbone and D. Ziou. On the Behavior of the Laplacian of Gaussian for Junction Models. In *Second Annual Joint Conference on Information Sciences*, pages 304–307, NC, USA, 1995.

[113] H.D. Tagare and R.J.P. deFigueiredo. Replay to "On Optimal Infinite Impulse Response Edge Detection Filters". IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 16(1):108–110, 1994.

[114] V. Torre and T.A. Poggio. On Edge Detection. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 8(2):147–163, Mar 1986.

[115] S. Venkatesh and L. J. Kitchen. Edge Evaluation Using Necessary Components. *CVGIP: Graphical Models and Image Processing*, 54(1):23–30, 1992.

[116] S. Venkatesh and R.A. Owens. On the Classification of Image Features. *Pattern Recognition*, 11:339–349, 1990.

[117] J.A. Webb and E. Pervin. The Shape of Subjective Contours. In *AAAI*, pages 340–343, 1984.

[118] D.J. Williams and M. Shah. Edge Contours Using Multiple Scales. *Computer Vision, Graphics and Image Processing*, 51:256–274, 1990.

[119] D.J. Williams and M. Shah. Edge Characterization Using Normalized Edge Detector. *Computer Vision, Graphics and Image Processing*, 55:311–318, 1993.

[120] A. P. Witkin. Scale-Space Filtering. In *Proceedings of the* 4^{th} *International Joint Conference on Artificial Intelligence*, pages 1019–1022, 1983.

[121] L. Wu and Z. Xie. Scaling Theorems for Zero-Crossings. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 12(1):46–54, Jan 1990.

[122] A.P Yuille and T.A. Poggio. Scaling Theorems for Zero-Crossings. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 8(1):15–25, June 1986.

[123] P. Zamperoni. Image Enhancement. In *Advances in Imaging and Electron Physics*, pages 1–76. San Diego : Academic Press, 1995.

[124] D. Ziou. Line Detection Using an Optimal IIR Filter. *Pattern Recognition*, 24(6):465–478, 1991.

[125] D. Ziou and J. P. Fabre. Effects of Edge Orientation on the Performances of First Order Operators. *Pattern Recognition Letters*, 15:1053–1063, 1994.

[126] D. Ziou and A. Koukam. The Selection of Edge Detectors Using Local Image Structure. In 7^{th} *IEEE International Conference on Tools with Artificial Intelligence*, pages 366–370, Virginia, Usa, 1995.

[127] D. Ziou and S. Tabbone. A Multi-Scale Edge Detector. *Pattern Recognition*, 26(9):1305–1314, 1993.

[128] D. Ziou and S. Tabbone. Adaptative Elimination of False Edges for First Order Detectors. In *Lecture Notes in Computer Science (Proceedings of the* 8^{th} *International Conference, Image Analysis and Processing), Vol. 974*, pages 89–94, Italy, 1995.

[129] D. Ziou and S. Wang. Isotropic Processing for Gradient Estimation. In *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*, pages 660–665, San Francisco, 1996.