LIDS-P-2075

Detecting and localizing edges composed of steps, peaks and roofs*

Pietro Perona† *and* **Jitendra Malik**‡

†Dipartimento di Elettronica ed Informatica
Università di Padova

†Center for Intelligent Control Systems,
MIT, Rm. 35-312, Cambridge MA 02139.
e-mail: perona@lids.mit.edu

‡Computer Science Division,
University of California at Berkeley, CA 94 720.
e-mail: malik@ernie.berkeley.edu

Abstract

It is well known that the projection of depth or orientation discontinuities in a physical scene results in image intensity edges which are not ideal step edges but are more typically a combination of steps, peak and roof profiles. However most edge detection schemes ignore the composite nature of these edges, resulting in systematic errors in detection and localization. We address the problem of detecting and localizing these edges, while at the same time also solving the problem of false responses in smoothly shaded regions with constant gradient of the image brightness. We show that a class of nonlinear filters, known as quadratic filters are appropriate for this task, while linear filters are not. A series of performance criteria are derived for characterizing the SNR, localization and multiple responses of these filters in a manner analogous to Canny's criteria for linear filters. A two–dimensional version of the approach is developed which has the property of being able to represent multiple edges at the same location and determine the orientation of each to any desired precision. This permits to recover edge junctions, and to calculate the orientation and curvature of the edges at each point. Experimental results are presented.

This research was supported by the Army Research Office under grant ARO DAAL-86-K-0171 (Center for Intelligent Control Systems).

1 Introduction

The problem of detecting and localizing discontinuities in greyscale intensity images has traditionally been approached as one of finding step edges. This is true both for the classical linear filtering approaches as well as the more recent approaches based on surface reconstruction.

Step edges are not always an adequate model for the discontinuities in the image that result from the projection of depth or orientation discontinuities in physical scene. Mutual illumination and specularities are quite common and their effects are particularly significant in the neighborhood of convex or concave object edges. In addition, there will often be a shading gradient on the image regions bordering the edge. As a consequence of these effects, real image edges are not step functions but more typically resemble a combination of of step, peak and roof profiles (Figure 1). This had been noted experimentally by Herskovits and Binford back in 1970. Quantitative analyses of the associated physical phenomena have also been provided- Horn[9] and more recently Forsyth and Zisserman [6].

The aim of this paper is to address the computational problem of detecting and localizing these composite edges. We propose to model these composite edges as arbitrary linear combinations of delta functions (modelling the narrow symmetric peaks due to specularities and mutual illumination), steps (modelling brightness discontinuities) and ramps (modelling the nonzero gradient component at the edges). The essential features of this model are (a) the fact that it is a multidimensional edge modelThe particular shape of these functions is somehwat irrelevant in the calculations. Blurred steps, instead of steps, and negative exponential profiles, instead of the delta functions, could have been chosen instead for an even more realistic modelling of the situation. The We restrict our attention to the *local* problem foregoing the use of 'context' or global information.

We begin by reviewing some related work in section 2. Most local edge detection methods are based on some decision-making stage following a linear filtering stage. Typically one looks for maxima in the filtered image perpendicular to the orientation of the edge. It is natural therefore to examine if that paradigm could be applied to the problem of detecting and localizing composite edges. In section 3 we show that this approach always leads to a systematic localization error for composite edges. Using any (finite) number of linear filters does not help. However, a *quadratic* filtering approach is adequate. Instead of looking for maxima in $(I * f)$ one looks for maxima in $(I * f_1)^2 + (I * f_2)^2$, or more generally $\sum (I * f_i)^2$. If one is to design an 'optimal' quadratic filtering approach, one needs to formulate computable forms of design criteria, analogous to the ones used by Canny [5] for linear filtering. We do this in Section 4. In the same section we examine the problem of false responses in the presence of smooth shading. We present conditions that are necessary and sufficient for a filter which does not suffer from this problem.

To detect edges in 2D, we use a Gaussian window to compute the 2D extension of the filter. A continuum of rotated copies of the filter is used to (conceptually) compute $R(x, y, \theta)$, a response that is a function of position and orientation. At each point, the locally dominant orientations θ_i which correspond to the local maxima of$R$ (over θ) are determined, and an edge is associated to each. Edge points are defined as the points where the directional derivative in the direction perpendicular to a locally dominant orientation is 0. Junctions are the points where multiple orientations are defined and two edges cross. Experimental results are presented.

2 Some relevant past work

Our work makes considerable use of insights and techniques due to Canny, and to Morrone, Owens and their colleagues. Here we review this work to provide a context.

Canny [5] studied the problem of designing 'optimal' linear shift–invariant filters for detecting and localizing a variety of image features: lines, steps and roofs. Initially, this problem is studied for isolated one–dimensional features bathed in white gaussian noise. He used three criteria to evaluate each operator

1. Good detection which is equivalent to maximizing the signal-to-noise ratio in the filter output,
2. Good localization which corresponds to ensuring that the the spatial location of the maximum of the output of the filter has a small variance

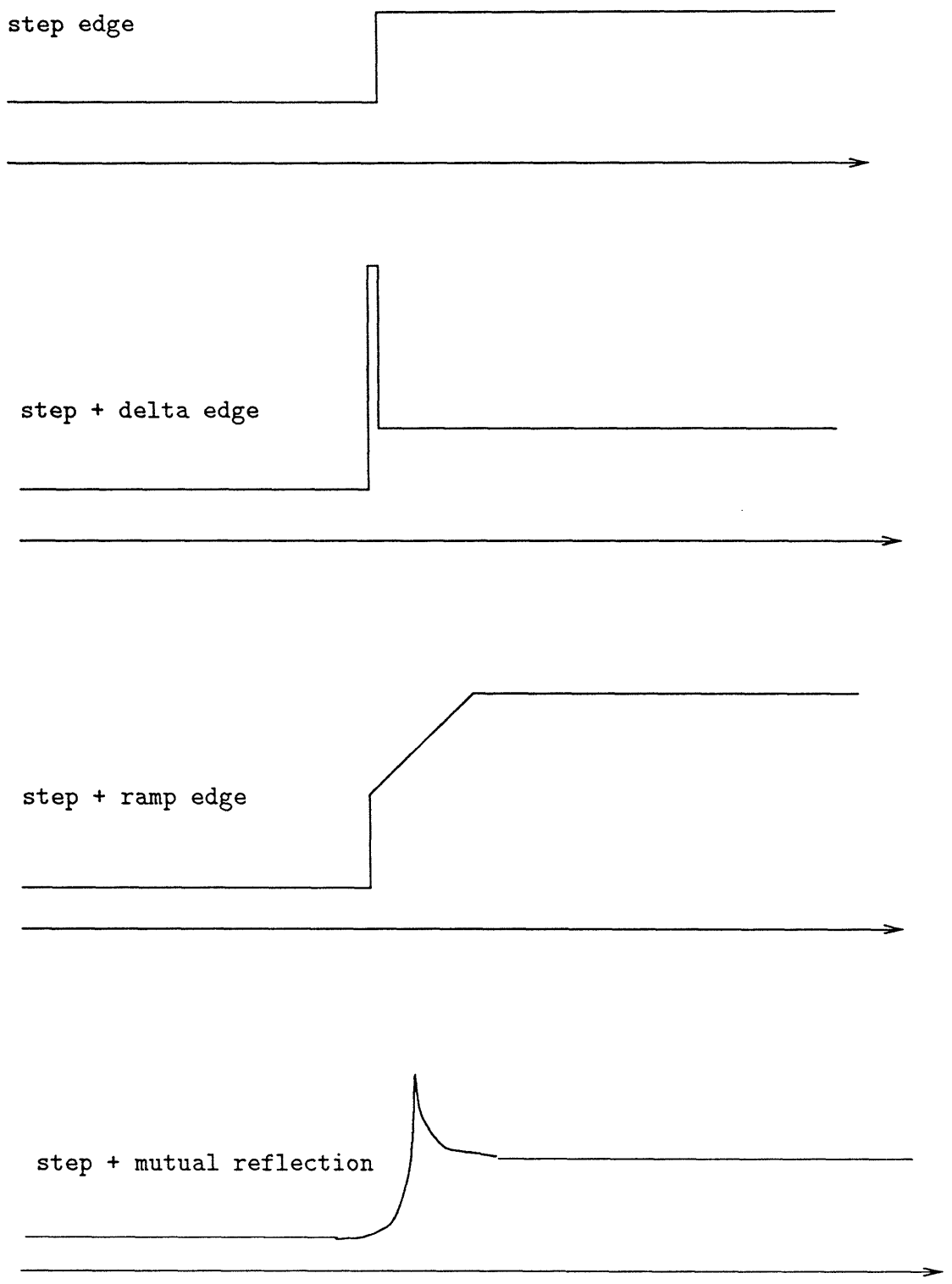

Figure 1: Some examples of edges.

3. Minimizing multiple responses to a single feature

By inspecting the shapes of the optimal filters, he concluded that near–optimal performance can be obtained by using the first derivative of a Gaussian G'_σ to detect steps, and the second derivative of a Gaussian G''_σ to detect lines and roofs. He used this 1D analysis as the basis of a 2D edge detection scheme, which incorporates some additional ideas–non-maximum suppression, thresholding with hysteresis, and feature synthesis to combine outputs of filters at multiple scales. The empirical performance of the Canny edge detector is good–it has become a widely–used piece of software, and a standard against which other edge detection schemes are often compared.

Canny's scheme suffers from three widely acknowledged problems, the first two having to do with it's 1D formulation:

1. False responses in smoothly shaded areas. Consider an area of the image where the gradient of the brightness is constant, but non–zero. No edges should be declared in such a region. However a first derivative operator would declare step edges whenever the magnitude of the gradient is above a threshold.

2. Incorrect localization of composite edges. The Canny edge detector has a systematic error in localization whenever there is a composite edge. For example, if the edge is a composite of a step and a roof the maxima of $I * G'_\sigma$ are at $x_\sigma = \sigma^2(k_2 - k_1)/h$ which is at the origin only if $k_1 = k_2$ [16](page 9). Therefore there is a systematic error for any non-zero value of σ whenever the two ramps have differing slopes k_1, k_2. Such asymmetric edges are reasonably common in real images in the presence of shading gradients in the two regions bounding the edge.

3. Loss of the junctions. The kernels employed by Canny have poor orientation selectivity; they admit only one 'dominant' orientation at each point. This results in a loss of the weaker edges at any junction (see Fig. (6) and the analysis by Beymer [2]).

The other piece of research relevant to this paper is that due to Morrone, Burr, Owens and Ross. Morrone et al [11] demonstrated by a series of psychophysical experiments that the human visual system detects features at points of strong *phase congruency*–these could be edges (spectral components have 0 phase), narrow bars (spectral components have 90 phase) or points on trapezoids where ramps meet plateaus (spectral components have 45 or 135 phase). To detect points of phase congruency, Morrone and Owens [13] find maxima of a local energy function $E(x) = F^2(x) + H^2(x)$ where $F(x)$ is the result of a convolution $I * f(x)$, and $H(x)$ is its Hilbert transform (equivalently I could be convolved with the Hilbert transform of f). The idea of using a quadrature pair of filters in this way actually goes back to Granlund's work on hierarchical picture processing [8], and an earlier model of human motion perception due to Adelson and Bergen [1]. Morrone and Owens show good empirical results for a particular choice of f. In later papers, Owens, Venkatesh and Ross [14] have shown that these operators are projections, and Morrone and Burr [12] have presented a more elaborated model of feature detection in human vision with four specific pairs of filters with parameters motivated by psychophysical data.

Their theory of human edge detection has several appealing properties – what attracts us most is the fact that it permits the detection of two different kinds of features in the same framework. If the problem is approached with a computational eye a few questions come to mind:

1. Is it necessary to use Hilbert pairs? Would any pair of odd–symmetric and even–symmetric filters work? Can these filters have different scales?

2. What kernels should one use?

3. Does such an apporach necessarily suffer from systematic localization errors as Canny's.

4. What is the performance in high brightness gradient areas?

5. Does the energy operator give a single maximum to a single feature?

3 Dealing with composite edges

We want to detect and localize edges which are arbitrary combinations of lines, steps and roofs. In this section we assume that the composite edge is $I = c_1\delta + c_2\delta^{(-1)}$. Ramps will come into the picture in Sec. 4.

A word about notation: we will write $f^{(-1)}(x)$ for $\int_{-\infty}^{x} f(t)dt$, and $f^{(-n)}(x) = (f^{(-n+1)})^{(-1)}(x)$. So $\delta^{(-1)}$ will be the step function and $\delta^{(-2)}$ a ramp.

First we establish an intuitive proposition which show that edge localization by looking at peaks in the responses of a fixed, finite family of linear filters leads to systematic errors.

Proposition 1 *For any fixed finite family of filters $\{f_1, f_2, \ldots, f_k\}$, there exists an image $I = c_1\delta + c_2\delta^{(-1)}$ for which none of the filter responses have a maximum at $x = 0$*

Proof. Edges are declared at the maxima of the response $I * f(x) = c_1 f(x) + c_2 f^{(-1)}(x)$. To ensure correct localization, there should be a maximum at $x = 0$ for any combination of c_1, c_2. For a filter f_i, its response has a maximum at $x = 0$ only if $(I * f_i)'(0) = 0$. Now $(I * f_i)' = c_1 f' + c_2 f$, implying that the vector $[c_1\ c_2]^T$ is orthogonal to $[f_i'(0)\ f_i(0)]^T$. To establish the proposition, one has only to pick a composite edge for which the vector $[c_1\ c_2]^T$ is not orthogonal to any of the vectors in the fixed, finite family of the k 2D vectors $[f_i'(0)\ f_i(0)]^T, i = 1, \ldots, k$.

In other words, if we had available to us the outputs of k different filters with a clever strategy which would enable us to pick the 'right' filter f_i whose response should be used to localize the edge, we would still be unable to guarantee zero localization error.

The problem is that for any particular linear filter we are able to construct a composite edge for which the filter is not matched. This suggests an alternative view–construct a parametrized filter which is a linear combination of an even filter f_e (matched to $\delta(x)$) and an odd filter f_o (matched to $\delta^{(-1)}$) and try to 'adapt' it to the particular composite edge in the image by picking the parameter value that maximizes the filter response at each point.

Call $f_\alpha(x) = \cos\alpha f_e(x) + \sin\alpha f_o(x)$ the filter, $I = c_1\delta + c_2\delta^{(-1)}$ the image, and $U(\alpha, x) = (I * f_\alpha)(x)$ the response. We want to choose α such that at each point x the response is maximized. Define $V(x) = \max_\alpha U(\alpha, x)$ and call $\alpha(x)$ the maximizing parameter (i.e. $V(x) = U(\alpha(x), x)$). Notice that $\alpha(x)$ must satisfy the equation $\frac{\partial}{\partial\alpha}U(\alpha(x), x) = 0$.

We would like the 'maximal' response $V(x)$ to have a maximum in zero, corresponding to the location of the edge: $V'(0) = (U_\alpha\alpha_x + U_x)(\alpha(0), 0) = 0$. Since $U_\alpha(\alpha(x), x) = 0$ then it must be $U_x(\alpha(0), 0) = 0$. Making use of the fact that $f_o(0) = f_e^{(-1)}(0) = 0$ we get the following system of equations:

$$U_x(\alpha(0), 0) = c_1 \sin\alpha f_o'(0) + c_2 \cos\alpha f_e(0) = 0 \tag{1}$$

$$U_\alpha(\alpha(0), 0) = -c_1 \sin\alpha f_e(0) + c_2 \cos\alpha f_o^{(-1)}(0) = 0 \tag{2}$$

The maximizing value of α, $\alpha(0)$, can be obtained from Equation 2. Substituting this into Equation 1 gives the following condition:

$$f_e^2(0) = -f_o^{(-1)}(0) f_o'(0) \tag{3}$$

If this condition is satisfied (see Sec. 4.5 for a more general formulation), the mixed edge $c_1\delta + c_2\delta^{(-1)}$ will be localized exactly by the maximum of $V(x)$ defined above.

An alternative approach yields the same condition. Define the vector of filters $F(x) = [f_e(x), f_o(x)]^T$. We localize features by looking for local maxima in the norm of the (vector) response to this filter of I. The squared norm of the response, $|\ I * F\ |^2$ is

$$W(x) = \{c_1\delta + c_2\delta^{(-1)} * f_e\}^2 + \{c_1\delta + c_2\delta^{(-1)} * f_o\}^2 \tag{4}$$

Equating the derivative of this expression with respect to x at the origin to 0 gives the condition

$$c_1 c_2 f_e^2(0) - c_1 c_2 f_o'(0) f_o^{(-1)}(0) = 0 \tag{5}$$

which is the same as Equation 3. Notice that given two kernels f_e and f_o it is possible to satisfy (3) by scaling one with respect to the other appropriately.

In conclusion, we have the possibility of getting arbitrarily precise localization of composite edges simply by looking for peaks in the response to a *quadratic* filter, i.e. in $\sum(I * f_i)^2$. Notice that the above condition only provides a normalization condition on the kernels employed, while their shape is not constrained. To make a proper choice of these kernels, one should instead bring to bear the criteria of having a good signal-to-noise ratio, low stochastic localization error etc. analogous to the approach used by Canny for linear filters.

4 Computation of the performance criteria

In the choice of a filter one would like to minimize different types of edge-detection errors. What follows is a list of criteria for evaluating quadratic filtering-based edge-detectors.

SNR – Signal to noise ratio (Sec. 4.2) – Ratio of signal response to the variance of the response due to noise.

DM – Multiple edges (Sec. 4.3) Edges not present in the signal found in the neighbourhood of real edges, due to multiple ripples in the convolution kernels.

DF – Spurious edges (Sec. 4.4) Edges detected far from 'true' edges. Due to response to high brightness gradient regions.

DL – Localization error (Sec. 4.5) – Error committed in locating the edge in the no-noise situation.

SL – Localization error (Sec. 4.6) – Localization error due to noise.

SM – Multiple responses (Sec. 4.7) – Edges detected in the neighbourhood of a true one due to noise in the data.

SF – Multiple responses (Sec. 4.8) – Edges detected far from the 'true' edges. Due to response to noise.

After establishing some notation this section is devoted to making a quantitative assessment of these criteria.

4.1 Notation

Edge – $G_c(x) = c_1\delta^{(-2)}(x) + c_2\delta^{(-1)}(x) + c_3\delta(x)$, $\mathbf{c}^T = [c_1, c_2, c_3]$

Noise – $N(x) = n_0\eta(x)$, $\eta(x)$ being white zero-mean unit-variance Gaussian noise.

Image – $I(x) = G_c(x) + N(x)$ - Signal + noise.

Kernels – $\mathbf{f}(x)^T = [f_1(x), \ldots, f_n(x)]$, and, for convenience, $\mathbf{F}(x)^T = [F(x)_1, \ldots, F(x)_n]$, with $\mathbf{F}''(x) = \mathbf{f}(x)$

Responses – $\mathbf{r}_{G_c}(x) = (\mathbf{f} * G)(x)$; $\quad \mathbf{r}_N(x) = (\mathbf{f} * N)(x) = n_0\mathbf{r}_\eta(x)$; $\quad \mathbf{r}(x) = \mathbf{r}_{G_c}(x) + \mathbf{r}_N(x)$

Power – $W_{G_c}(x) = \|\mathbf{r}_{G_c}(x)\|^2$; $\quad W(x) = \|\mathbf{r}(x)\|^2$

Correlations – The nxn correlation matrix $\mathbf{R}_{fg}(t)$ is defined componentwise by:

$$\mathbf{R}_{fg(ij)}(t) \doteq (f_i(\cdot + t), g_j(\cdot))_{L_2}$$

For simplicity of notation whenever $f = g$ we will write $\mathbf{R}_f$ instead of $\mathbf{R}_{ff}$.

Kernel derivatives – $\mathbf{H}$ defined componentwise by $\mathbf{H}_{ij} = F_i^{(j-1)}$, $j = 1, \ldots, 3$ and $i = 1, \ldots, n$.

4.2 Signal to noise ratio

Define signal to noise ratio as the ratio of the response to pure signal at the edge and the standard deviation of the response to pure noise. Using our notation:

$$SNR \doteq \frac{\|\mathbf{r}_{G_c}(0)\|}{\sqrt{E\|\mathbf{r}_N\|^2}} \tag{6}$$

The variance of the response to pure noise depends on the correlation matrix of the functions f(x):

$$E\|\mathbf{r}_N\|^2 = E\sum_i (f_i * N)^2 = n_0^2 \sum_i \langle f_i, f_i \rangle = tr(\mathbf{R}(0)) \tag{7}$$

For a generic edge image $G_c(x)$ the signal to noise ratio is then:

$$SNR = \frac{\|\mathbf{r}_{G_c}(0)\|}{n_0\sqrt{tr(\mathbf{R}(0))}} \tag{8}$$

In the special case that the edge is a combination of roof, step and line: $G = c_1\delta^{(-2)} + c_2\delta^{(-1)} + c_3\delta$ the signal to noise ratio becomes (see Section A.1):

$$SNR = \frac{\|\mathbf{Hc}\|}{n_0\sqrt{tr(\mathbf{R}(0))}} \tag{9}$$

4.3 Ripples in the filters

Our strategy for detecting edges is to look for maxima in the filter response. It is important therefore that the resposnse of the filters in the no-noise situation, $W(x) = W_{G_c}(x)$, has only one peak per edge. Therefore a requirement that we make is that $W_{G_c}(x)$ is unimodal, independently of the composition of the edge, i.e. that (call x_e the coordinate of the maximum)

$$\frac{\partial}{\partial x} W_{G_c}(x) > 0 \qquad x < x_e \qquad \forall c \tag{10}$$

$$\frac{\partial}{\partial x} W_{G_c}(x) < 0 \qquad x > x_e \qquad \forall c \tag{11}$$

Since $W_{G_c} = \|\mathbf{Hc}\|^2$ the condition is equivalent to the matrix $\mathbf{H'}^T\mathbf{H}$ being positive definite for $x < x_e$ and negative definite for $x > x_e$ (cfr the condition that the matrix is antisymmetric in zero, sec. 4.5).

4.4 Dealing with shading gradients

A well known problem of first derivative edge detectors is that they respond with false edges in areas with smooth shading even when the gradient of brightness is constant. To avoid these false positives, one may have to set a threshold which leads to the rejection of genuine low–contrast edges. This problem has persisted in the 'modern' approaches based on surface reconstruction. Whether the formulation is a probabilistic one using MRFs (e.g. Geman and Geman) or a variational one, if the cost function includes terms like the squared gradient there will be a tendency towards piecewise constant reconstructions. Blake and Zisserman [4] even provide a computation of the gradient threshold above which false edges get created.

In the linear filtering framework, Binford [3] describing the Binford–Horn line finder discusses one solution to this problem– a lateral inhibition stage preceding the stage of finding directional derivatives. Essentially this amounts to using third derivatives, and suffers from the expected weakness–low signal to noise ratio compared to first derivative operators. A simple calculation using the SNR criterion defined by Canny [5] confirms this.

A compact characterization of filters which do not suffer from the linear gradient problem can be obtained as follows: suppose that the image just consists of a ramp function $I(x) = \delta^{(-2)}(x)$. The response of a linear filter f to such a ramp is $I * f = f^{(-2)}(x)$. It can be seen that $f^{(-2)}(x)$ should satisfy the following two conditions:

1. $\|f^{(-2)}(x)\| \to 0$ for $\|x\| \to \infty$. This ensures that far enough from the roof junction, the response to a ramp is negligible.

2. $f^{(-2)}(x)$ either has a zero crossing or a maximum or a minimum at the origin. This is to enable the localization of onset of the ramp without any bias.

While the third derivative of a Gaussian $G'''_\sigma(x)$ is one filter which would satisfy these criteria, there are others which do so without that significant a drop in SNR. One such choice is the Hilbert Transform of $G''_\sigma(x)$ which is an odd-symmetric filter. We computed Canny's SNR and localization criteria for this filter and compared it with $G'_\sigma(x)$. It turns out that for $G'_\sigma(x)$, the SNR is $1.062\sigma^{0.5}$ and localization is $0.8673\sigma^{-0.5}$. For $(G''_\sigma)_H(x)$, the SNR is $0.6920\sigma^{0.5}$ and localization is $0.87535\sigma^{-0.5}$. Considering the product of the SNR and the localization, the numbers are 0.92 and 0.606 respectively implying that $(G''_\sigma)_H$ is worse by about 34%. However, its r value is 0.676 which is 32 % better than $r = 0.51$ for the G'_σ. In other words, while the $(G''_\sigma)_H$ is roughly comparable to the G'_σ filter used by Canny, its immunity to smooth shading makes it preferable.

This observation is independent of our other main concern in this paper–that of correctly dealing with composite edges–and could be directly exploited. One caveat: Since this filter has additional side lobes, the reponse to an ideal step edge shows a ringing like phenomenon. These spurious side maxima need to be suppressed, which could be done by using a locally adaptive threshold along the lines of Malik and Perona[10]. However in this paper, where we use such a filter as part of a quadratic filter, this suppression is not necessary.

For a particular choice of quadratic filter, namely $f_e = G''_\sigma$ and $f_o = (G''_\sigma)_H$, Figures 2 and 3 show the response to a number of different stimuli. Note how in each case, the composite edge is correctly localized and that the filter is insensitive to linear shading.

4.5 DL – Systematic localization error – Normalization conditions

In the scheme we propose edges are detected whenever $W(x)$ reaches a maximum. A natural and simple criterion for localizing the position of each detected edge is to define it to be the coordinate $x = x_e$ where W reaches the maximum. This localization criterion is quite arbitrary: given a generically shaped edge signal it is not clear where the edge should be located. In some special cases, however, the position of the boundaries is naturally defined. Consider a signal G defined as in section 4.1; whatefer the choice of the coefficients $\mathbf{c}$, the edge is located at $x = 0$, so $x_e - 0 = x_e$ is a localization error. In this section x_e is computed for a given choice of twice continuously differentiable filters $\mathbf{f}$ when the noise is zero, i.e. $W = W_{G_c}$ (in the rest of this section we will write W instead of W_{G_c}).

A necessary and sufficient condition for x_e to be a maximum point is that $W'(x_e) = 0$ and $W''(x_e) < 0$. Expanding W' in Taylor sum around $x = 0$ and computing it in $x = x_e$ we obtain:

$$0 = W'(x_e) = W'(0) + W''(0)x_e + \mathcal{O}(x_e^2) \tag{12}$$

which gives us an estimate of x_e in terms of the derivatives of W at the origin:

$$x_e \approx \frac{W'(0)}{-W''(0)} \tag{13}$$

The derivatives of W are computed in section A.1:

$$x_e \approx -\frac{\mathbf{c}^T\mathbf{H}'^T(0)\mathbf{H}(0)\mathbf{c}}{\mathbf{c}^T\mathbf{H}''^T(0)\mathbf{H}(0)\mathbf{c} + \|\mathbf{H}'(0)\mathbf{c}\|^2} \tag{14}$$

A necessary and sufficient condition for the systematic localization error to be zero is therefore that the filter collection $\mathbf{f}$ satisfies the conditions:

$$\mathbf{c}^T\mathbf{H}'^T(0)\mathbf{H}(0)\mathbf{c} = 0 \qquad \forall \mathbf{c} \in \mathbb{R}^3 \tag{15}$$

$$\mathbf{c}^T\mathbf{H}''^T(0)\mathbf{H}(0)\mathbf{c} < -\|\mathbf{H}'(0)\mathbf{c}\|^2 \qquad \forall \mathbf{c} \in \mathbb{R}^3 \tag{16}$$

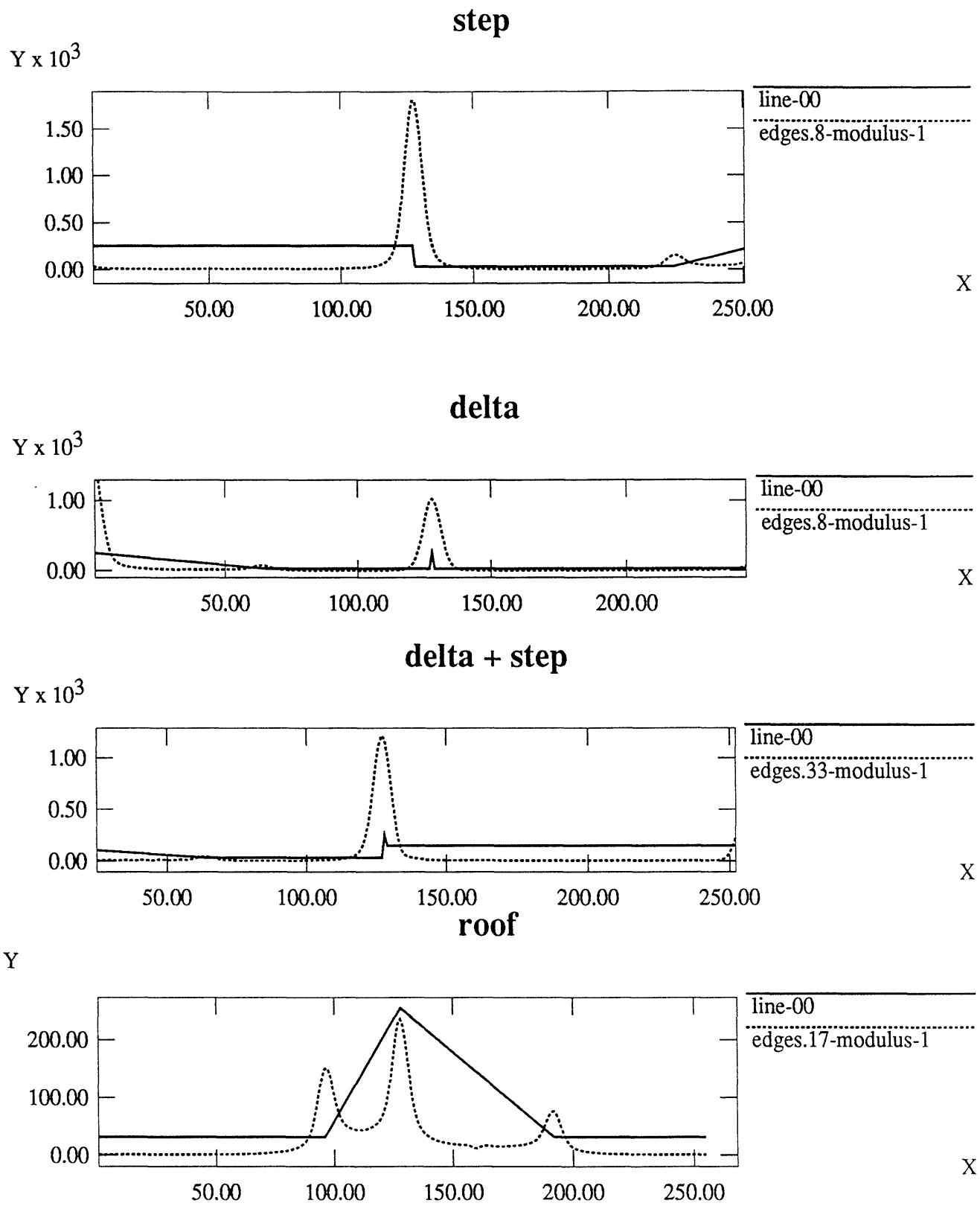

Figure 2: 1 dimensional examples. The energy peaks correspond to the edge position and the constant gradient areas generate zero energy.

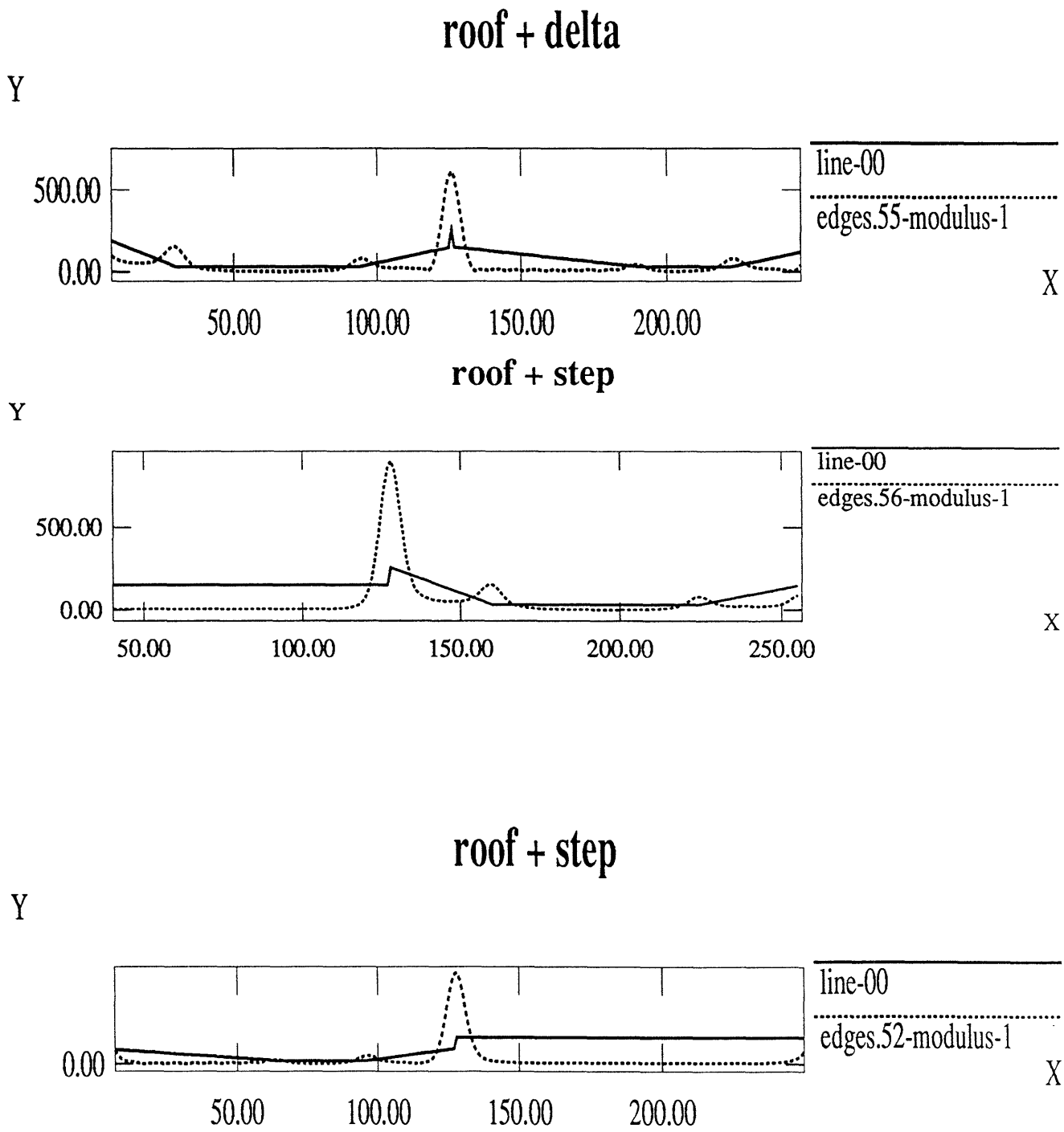

Figure 3: 1 dimensional examples. The energy peaks correspond to the edge position and the constant gradient areas generate zero energy.

Notice that condition (15) is equivalent to $\mathbf{H}'^T(0)\mathbf{H}(0)$ being an antisymmetric matrix. In fact any matrix $\mathbf{A}$ may be written as the sum of its symmetric and antisymmetric components $2\mathbf{A} = 2\mathbf{A}_s + 2\mathbf{A}_a \doteq (\mathbf{A}+\mathbf{A}^T)+(\mathbf{A}-\mathbf{A}^T)$. The product $\mathbf{c}^T\mathbf{A}\mathbf{c}$ is equal to the product involving the symmetric part only: $\mathbf{c}^T\mathbf{A}\mathbf{c} = \mathbf{c}^T\mathbf{A}_s\mathbf{c}$. The symmetric part may be diagonalized ($\mathbf{A}_s \doteq \mathbf{T}^T\mathbf{D}\mathbf{T}$) giving $0 = \mathbf{c}^T\mathbf{A}\mathbf{c} = (\mathbf{T}\mathbf{c})^T\mathbf{L}(\mathbf{T}\mathbf{c}) \Leftrightarrow \mathbf{L} = 0 \Leftrightarrow \mathbf{A}_s = 0 \Leftrightarrow \mathbf{A} = -\mathbf{A}^T$.

Equation (15) is then equivalent to:

$$\mathbf{H}'^T(0)\mathbf{H}(0) = -\mathbf{H}^T(0)\mathbf{H}'(0) \tag{17}$$

which may be written componentwise as:

$$\sum_k \mathbf{H}'_{k,i}\mathbf{H}_{k,j} = -\sum_k \mathbf{H}_{k,i}\mathbf{H}'_{k,j} \tag{18}$$

From the definition of $\mathbf{H}$:

$$\sum_{k=1}^{n} F_k^i(0)F_k^{(j-1)}(0) = -\sum_{k=1}^{n} F_k^j(0)F_k^{(i-1)}(0) \qquad i,j = 1\ldots 3 \tag{19}$$

Observe that when $\mathbf{f}^T = [f_1, f_2] = [f_e, f_o]$ with f_e and even kernels and f_o an odd kernel, as in section 3, the equations 19 specialize as follows: (a) when $i+j$ is even the equations are trivially satisfied since they are formed of products $F_i^r F_i^t$ with $s+t$ odd, and since F_e has odd derivatives equal to zero in zero and F_o has even derivatives equal to zero in zero all such products are zero. (b) when $i+j$ is odd then the equations are:

$$(i = 1, j = 2), (j = 1, i = 2) \qquad F'^2_2(0) = -F''_1(0)F_1(0) \tag{20}$$

$$(i = 2, j = 3), (j = 2, i = 3) \qquad F''^2_1(0) = F'''_2(0)F'_2(0) \tag{21}$$

Equation (21) is exactly Equation (3), while Equation (20) is added by the introduction of the ramp. In conclusion: when f_e and f_o are such that Eq. (21) and Eq. (3) are satisfied there will be no systematic localization error for any edge which is a linear combination of a step, delta and ramp. Two observations are of importance: (a) While Eq. (3) may be easily satisfied by renormalizing appropriately a previously chosen pair of kernels, the pair of equations (20) and (21) is in general more stringent. (b) The fact that we have used a ramp, step and delta to build our model of an edge somehwat simplifies the calculations, but it is not of great importance. For example an exponential $exp(-\tau||x||)$ could be used in place of the delta function, and the calculations would carry through in the same way.

4.6 SL – Stochastic localization error

In this section we study the localization error due to noise added to the image. We label $x = 0$ the position where the response $W_{G_c}(x)$ to the noiseless signal has a peak (i.e. $W'_{G_c}(0) = 0$), and $x = x_0$ the coordinate where the response to noisy signal $W(x)$ does (i.e. $W'(x_0) = 0$). We expand the derivative $W'(x)$ of the response in power series around the $(x = 0, n_0 = 0)$ point, and compute it at the (x_0, n_0) point.

The derivative of W in x_0 is:

$$W'(x_0) = W'_{G_c}(x_0) + 2n_0\left(\mathbf{r}^T_{G_c}\mathbf{r}_\eta\right)'(x_0) + \mathcal{O}(n_0^2) = 0 \tag{22}$$

The first two terms of the second member of equation (22) can be expanded in Taylor sums around the origin:

$$W'_{G_c}(x_0) = W'_{G_c}(0) + W''_{G_c}(0)x_0 + \mathcal{O}(x_0^2) = W''_{G_c}(0)x_0 + \mathcal{O}(x_0^2) \tag{23}$$

$$\left(\mathbf{r}^T_{G_c}\mathbf{r}_\eta(x_0)\right)' = \left(\mathbf{r}^T_{G_c}\mathbf{r}_\eta\right)'(0) + \left(\mathbf{r}^T_{G_c}\mathbf{r}_\eta\right)''(0)x_0 + \mathcal{O}(x_0^2) \tag{24}$$

Substituting equations (23), (24) into equation (22) and solving for x_0 we obtain:

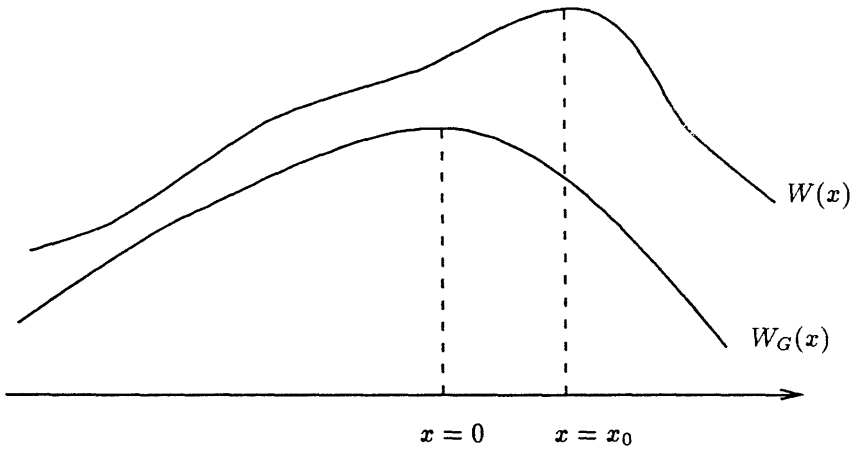

Figure 4: Localization error due to noise.

$$x_0 \approx n_0 \frac{2\left(\mathbf{r}_{G_c}^T \mathbf{r}_\eta\right)'(0)}{-W''_{G_c}(0) + \mathcal{O}(n_0)} \approx n_0 \frac{2\mathbf{r}'^T_{G_c}\mathbf{r}_\eta(0) + 2\mathbf{r}^T_{G_c}\mathbf{r}'_\eta(0)}{-W''_{G_c}(0)} \tag{25}$$

The expectation of the stochastic localization error is clearly zero, and the variance may be approximated using (25):

$$Ex_0^2 \approx n_0^2 \frac{E(\mathbf{r}_{G_c}^T \mathbf{r}_\eta(0))^2}{W''_{G_c}(0)^2} = n_0^2 \frac{\mathbf{r}'^T_{G_c}\mathbf{R}_f\mathbf{r}'_{G_c}(0) + 2\mathbf{r}'^T_{G_c}\mathbf{R}_{ff'}\mathbf{r}_{G_c}(0) + \mathbf{r}^T_{G_c}\mathbf{R}_{f''}\mathbf{r}_{G_c}(0)}{W''_{G_c}(0)^2} \tag{26}$$

4.7 SM – Spacing of the maxima in the neighbourhood of an edge

We will suppose that the noise variance, n_0, is small with respect to the magnitude of the signal. We will therefore approximate the value of $W(x)$ disregarding term quadratic in n_0:

$$W = W_{G_c} + 2n_0\mathbf{r}^T_{G_c}\mathbf{r}_\eta + n_0^2 W_\eta \tag{27}$$

$$\approx W_{G_c} + 2n_0\mathbf{r}^T_{G_c}\mathbf{r}_\eta \tag{28}$$

We may apply equation (87) which is a generalization of Rice's formula [18] to compute the expected value of the distance between maxima of the random process $W_\Delta(x) = W(x) - W_{G_c}(x) = 2n_0\mathbf{r}^T_{G_c}\mathbf{r}_\eta$. Notice that the expectation has an argument x since it depends on the distance from the location of the edge $G_c(x)$. In a neighbourhood of the edge we expect the derivative of W_G to be close to zero and thus the estimate of the spacing of the maxima of $W_\Delta(x)$ to be a good estimate of the spacing of the maxima of $W(x)$. For the sake of simplicity we evaluate the function in $x = 0$ (in principle one should verify that $x = 0$ is the worst case):

$$d_{W_\Delta}(0) = 2\pi \frac{\sqrt{a(0)c(0) - b^2(0)}}{c(0) - b(0)} \tag{29}$$

Where $a(x)$, $b(x)$ $c(x)$ are defined by equation (85) and in $x = 0$ simplify to:

$$\begin{aligned}
a(0) &= \mathbf{r}'_{G_c}(0)^T\mathbf{R}_f\mathbf{r}'_{G_c}(0) + \mathbf{r}_{G_c}(0)^T\mathbf{R}_{f'}\mathbf{r}_{G_c}(0) \\
b(0) &= \mathbf{r}'_{G_c}(0)^T\mathbf{R}_f\mathbf{r}''_{G_c}(0) + \mathbf{r}'_{G_c}(0)^T\mathbf{R}_{ff''}\mathbf{r}_{G_c}(0) + 2\mathbf{r}_{G_c}(0)^T\mathbf{R}_{f'}\mathbf{r}'_{G_c}(0) \\
c(0) &= \mathbf{r}''_{G_c}(0)^T\mathbf{R}_f\mathbf{r}''_{G_c}(0) + 4\mathbf{r}'_{G_c}(0)^T\mathbf{R}_{f'}\mathbf{r}'_{G_c}(0) + \mathbf{r}_{G_c}(0)^T\mathbf{R}_{f''}\mathbf{r}_{G_c} + 2\mathbf{r}''_{G_c}(0)^T\mathbf{R}_{ff''}\mathbf{r}_{G_c}(0)
\end{aligned} \tag{30}$$

4.8 SF – Spacing of the maxima far from an edge

We have not been able to compute this in closed form a function of the kernels' shape. One may take the sum of the estimates of the number of maxima of $f_k * \eta$ as an upper bound to this number, therefore, using Rice's formula (65):

$$T_{[1..n]} = 2\pi \sum_{k=1}^{n} \sqrt{\frac{R_{f_k'}(0)}{R''_{f_k'}(0)}} \tag{31}$$

5 Detecting edges in two dimensions

To detect edges in 2D, we use a Gaussian window to compute the 2D extension of the 1D kernels discussed so far. Rotated copies of the kernels are used to (conceptually) compute a response $R(x, y, \theta)$. At each point, the locally dominant orientations θ_i which correspond to the local maxima (over θ) are determined. Allowing for multiple orientations enables junctions to be detected consistently. Edge points are defined as the points where the directional derivative in the direction perpendicular to a locally dominant orientation is 0.

In practice one cannot afford to compute convolutions of the image with kernels at an infinity of orientations. It turns out that it is possible to approximate $R(x, y, \theta)$ with arbitraty precision using linear combinations of a finite number of functions as described in the next section. It is therefore possible to compute $R(x, y, \theta)$ on a continuum of orientations (see Fig.5-b).

5.1 Computing filter responses on a continuum of orientations

The problem is the following: Given the kernel $F : \mathbf{R}^2 \rightarrow \mathbf{C}$, define the family of 'rotated' copies of F as: $F_\theta = F \circ R_\theta$, $\theta \in \mathbb{S}^1$, where $\mathbb{S}^1$ is the circle and R_θ is a rotation. Is it possible to express F_θ as

$$F_\theta(\mathbf{x}) = F \circ R_\theta(\mathbf{x}) = \sum_{i=1}^{n} \beta(\theta)_i G_i(\mathbf{x}) \qquad \forall \theta \in \mathbb{S}^1, \forall \mathbf{x} \in \mathbf{R}^2 \tag{32}$$

a *finite* linear combination of functions $G_i : \mathbf{R}^2 \rightarrow \mathbf{C}$ with θ-dependent coefficients $\beta(\theta)$? If this is feasible then the convolution of the kernel F_θ with an image I is also a finite sum: once n convolutions $S_i = I * G_i$ have been computed, in order to obtain the filter response $S(x, y, \theta)$ at any orientation θ in a continuum we only need to calculate a linear combination of the finite S_i:

$$S_\theta(\mathbf{x}) = I * F \circ R_\theta = \sum_{i=1}^{n} \beta(\theta)_i I * G_i = \sum_{i=1}^{n} \beta(\theta)_i S_i \tag{33}$$

One has to be aware of the fact that for most functions F an exact finite decomposition of F_θ as in Eq. (32) cannot be found (see Freeman and Adelson [7] for examples of kernels that *can* be exactly expressed with a finite sum), however it is possible to compute good approximations with a small number of components. In fact, for the kernels of interest in this paper the approximation error decreases exponentially fast with the number of components in the approximation. The optimal n-approximation can be computed by considering the singular value decomposition of a linear operator associated with the kernel F_θ. We enunciate here a theorem that is demonstrated and discussed in [15], along with a more general statement and a discussion of some implementation details.

Consider first the approximation to F_θ defined as follows:

Definition. Call $F_\theta^{[n]}$ the n-terms sum:

$$F_\theta^{[n]} = \sum_{i=1}^{n} \sigma_i a_i(\mathbf{x}) b_i(\theta) \tag{34}$$

with σ_i, a_i and b_i defined in the following way: let $\hat{h}(\nu)$ be the (discrete) Fourier transform of the function $h(\theta)$ defined by:

$$h(\theta) = \int_{\mathbf{R}^2} F_\theta(\mathbf{x})\overline{F_{\theta'=0}(\mathbf{x})}d\mathbf{x} \tag{35}$$

and let ν_i be the frequences on which $\hat{h}(\nu)$ is defined, ordered in such a way that $\hat{h}(\nu_i) \geq \hat{h}(\nu_j)$ if $i \leq j$. Call $N \leq \infty$ the number of nonzero terms $\hat{h}(\nu_i)$. Set now:

$$\sigma_i = \hat{h}(\nu_i)^{1/2} \tag{36}$$

$$b_i(\theta) = e^{j2\pi\nu_i\theta} \tag{37}$$

$$a_i(\mathbf{x}) = \sigma_i^{-1} \int_{\mathbf{S}^1} \overline{F_\theta(\mathbf{x})} e^{j2\pi\nu_i\theta} d\theta \tag{38}$$

Then

Theorem 1 *$F_\theta^{[n]}$ is the best (in the sense that it minimizes the L^2-norm of the error) n-dimensional approximation to F_θ.*

In our experiments we used Gaussian second derivatives and their Hilbert transform. With fairly orientation-selective (approximately 30^o) kernels of ratios $\sigma_x : \sigma_y = 1 : 3$ a total of 9 filters was sufficient to keep the error around 10%.

5.2 Edge detection

At edge points the filter output 'energy' R (see Fig.5-b) will have a maximum at the orientation θ_e parallel to the edge (see Fig.5-c). Fix θ_e and consider $R(x, y, \theta_e)$. Along a line orthogonal to the edge the problem reduces to the 1D case: there will be an energy maximum at the edge. Points through which edges pass can be found by marking as 'edge points' all the points $p = (x, y, \theta)$ that satisfy:

$$\frac{\partial}{\partial\theta} R(p) = 0 \tag{39}$$

$$\frac{\partial}{\partial\mathbf{v}_\theta} R(p) = 0 \tag{40}$$

where $\mathbf{v}_\theta$ is the unit vector orthogonal to the orientation associated to θ.

The search for the edge points has been implemented as follows:

1. For each image pixel (x, y) the angles $\theta_i(x, y)$ at which the response is maximized are found. For this operation we use Brent's maximization algorithm (See [17]). The upper bound on the orientation error was set at 1 degree (see Fig.5-c). The angle space is coarsely sampled (approx. a sample every 5 degrees) to provide initial conditions for the bracketing algorithm. The energies $R_i(x, y)$ corresponding to $\theta_i(x, y)$ are also stored. The lower 70% of the sampled energies at each point are averaged to give a global noise estimate.

2. The maxima in (x, y) of $R(x, y, \theta_i(x, y))$ are computed with sub-pixel accuracy by fitting a cylindrical paraboloid to $R(x, y, \theta_i)$ in the 3x3 neighbourhood around the pixel (See Fig.5-d). Points where the fit error (see Eq. (51) is greater than a preset threshold value are discarded. A typical threshod we used was 15%. See section 5.3 for the details.

3. The edge pixels are thresholded. The thresholding is based on a threshold based on an estimate of the average noise.

The results of the search are shown in figures 7 and 5-d and compared to the output of a Canny detector using filters of the same scale. The typical number of steps of the Brent algorithm was 2-3. The time spent in the search was always less than the filtering time.

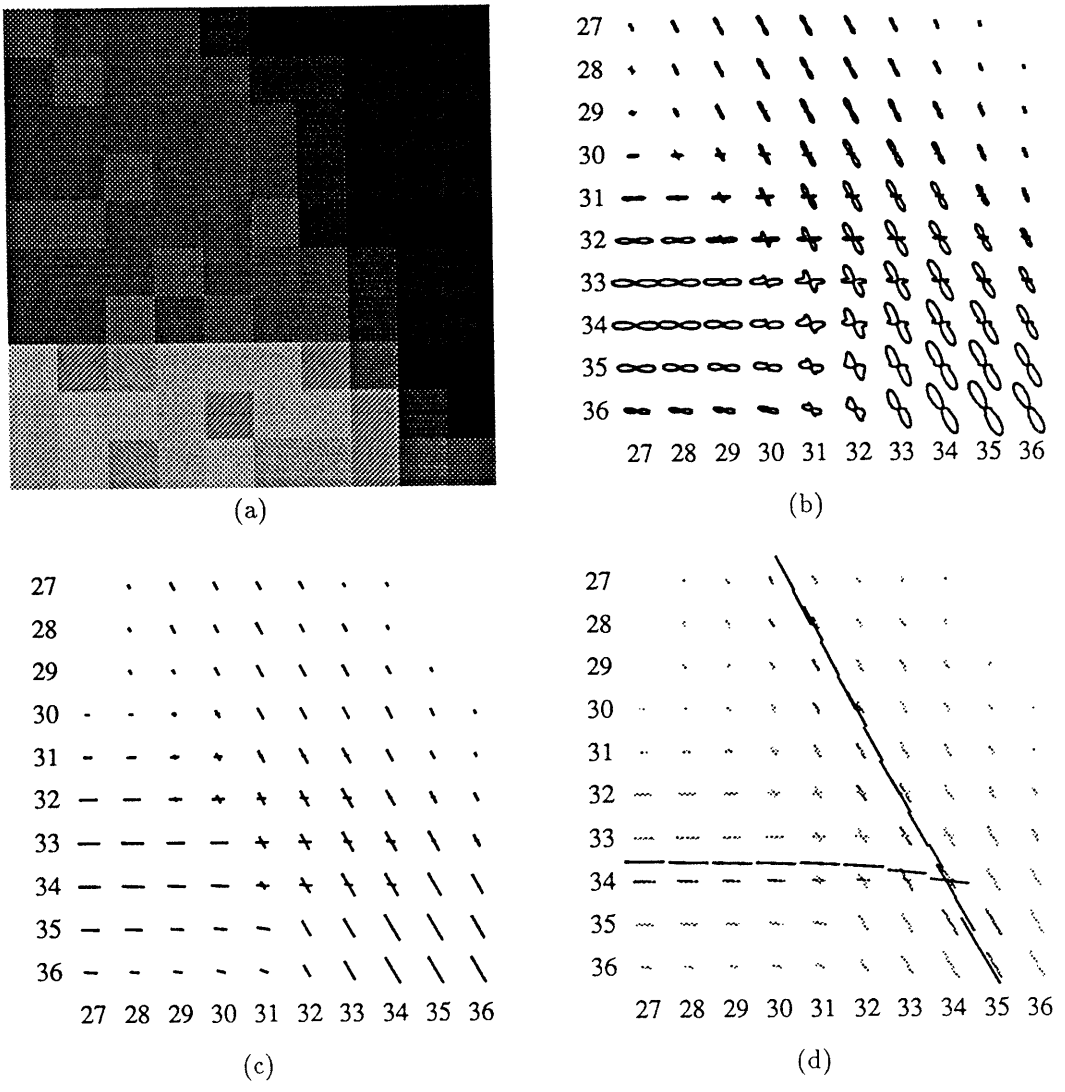

Figure 5: The process of brightness edge detection shown for a detail (a) of the T junction image of Fig. 6. (b) Energies are computed at every pixel as a function $R(x, y, \theta)$. At each pixel the energy is represented by a polar plot; the plots are π-periodic since $R(x, y, \theta)$ does not encode the direction, only the orientation of the edges. Notice that the energies have (c) Maxima in θ are calculated at each pixel. The length of the needles indicates the associated energy. (d) 'Oriented' maxima in (x, y) are computed at subpixel resolution.

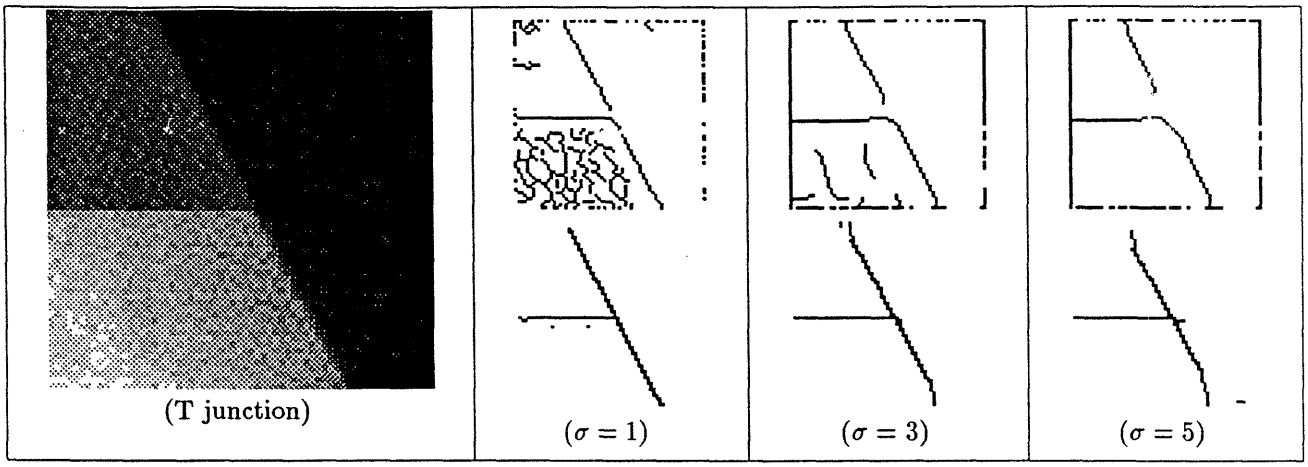

Figure 6: Comparison of the Canny detector (top) and our 2D detector (bottom) at three different scales. Notice the different performance at the junction: the Canny detector misses the junction (despite very low threshold settings) and deforms the remaining corner.

5.3 Sub-pixel localization of the edges

As we have discussed above, we associate the edges in the image with the 'ridges' of a function $W(x, y, \theta)$ (see Eq.(39). The W function is computed on the nodes of a lattice corresponding to the data lattice. This need not limit the accuracy with wich the position and the shape of the ridges can be determined. The function W has a very regular behaviour and may be interpolated to any point of a continuum in x, y, θ.

In this section we describe an interpolation method based on fitting a second order model to the 'dorsals' of the ridges of W at a certain angle θ. The model is a paraboloid whose axis of symmetry has an angle θ with the X axis, and is described by:

$$y = ad^2 + bd + c \qquad\qquad d(x, y) = -xsin\theta + ycos\theta \tag{41}$$

The model has three parameters a, b, c. The vector $[a, b, c]^T$ will be referred to as $\mathbf{a}$. Notice that the model is linear in $\mathbf{a}$.

Consider a 3x3 neighbourhood. The nine data points may be expressed in terms of the model in matrix form:

$$\mathbf{y} = A\mathbf{a} \qquad \mathbf{y}^T = [y_1, \ldots, y_9] \qquad A_{i,j} = d_i^j \qquad i = 1, \ldots, 9; j = 0, \ldots, 2 \tag{42}$$

with d defined as above.

The linear least squares estimator for $\mathbf{a}$ is then (see e.g. [17]):

$$\hat{\mathbf{a}} = (A^T A)^{-1} A^T \mathbf{y} \tag{43}$$

The matrix $(A^T A)^{-1}$ may be computed explicitly. Define the quantity $\beta = 2sin^2\theta cos^2\theta + 1$. Then:

$$A^T A = \begin{bmatrix} 6\beta & 0 & 6 \\ 0 & 6 & 0 \\ 6 & 0 & 9 \end{bmatrix} \tag{44}$$

and defining $\Delta = 6(3\beta - 2)$:

$$(A^T A)^{-1} = \Delta^{-1} \begin{bmatrix} 3 & 0 & -2 \\ 0 & 3\beta - 2 & 0 \\ -2 & 0 & 2\beta \end{bmatrix} \tag{45}$$

Therefore the operator associated to the estimator is:

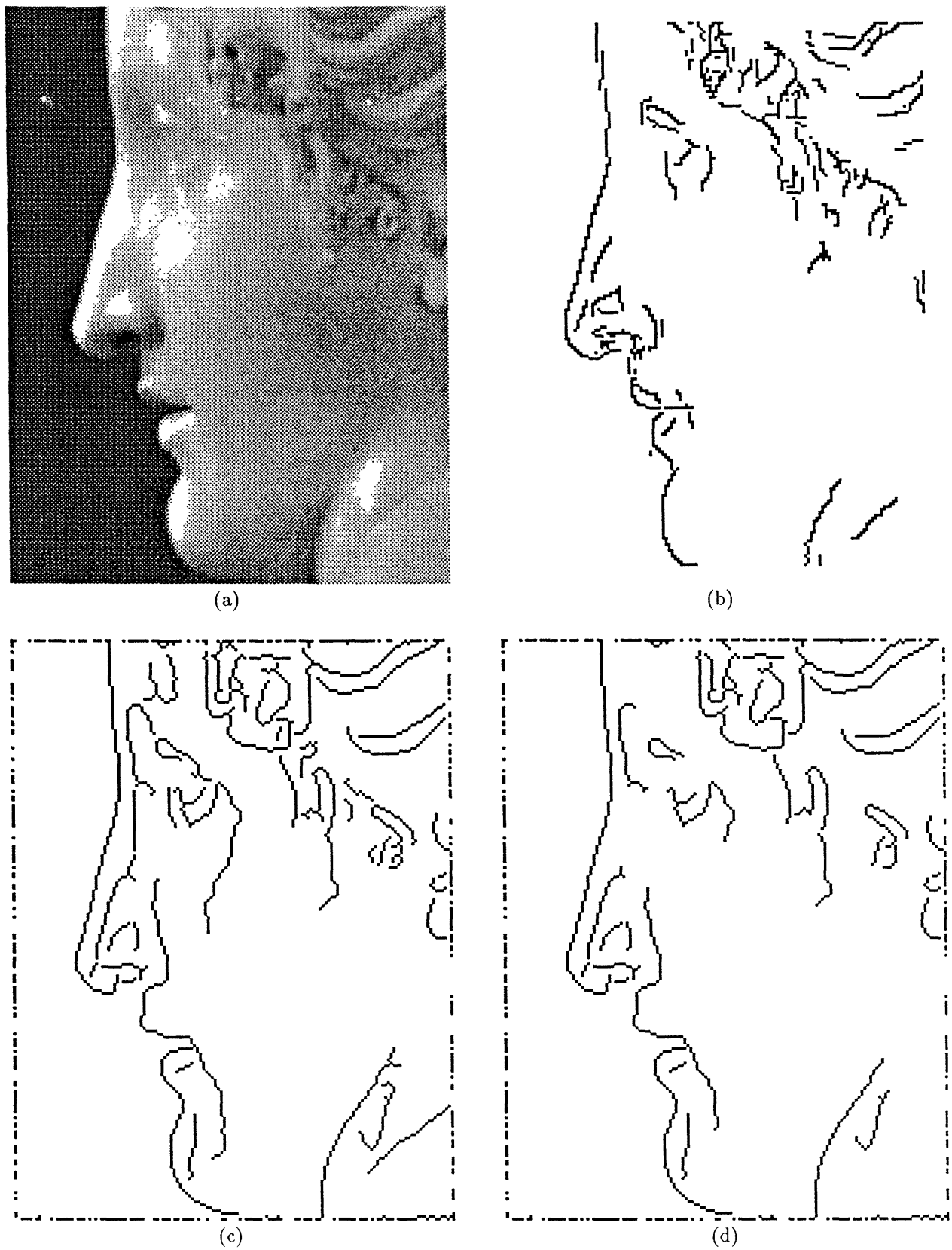

Figure 7: Comparison of the Canny detector and our 2D detector. (a) Original (Paolina Borghese, Canova circa 1800). (b) our detector, sigma = 1, ratio = 2. (c-d) Canny detector with sigma = 1, and thresholds (150,250), and (200,400) respectively.

$$(A^TA)^{-1}A^T = \Delta^{-1}\begin{bmatrix} 3d_1^2-2 & 3d_2^2-2 & \dots & 3d_9^2-2 \\ (3\beta-2)d_1 & (3\beta-2)d_2 & \dots & (3\beta-2)d_9 \\ -2d_1^2+2\beta & -2d_2^2+2\beta & \dots & -2d_9^2+2\beta \end{bmatrix} \quad (46)$$

So the parameters have the following closed-form estimators:

$$\hat{a} = \Delta^{-1}\sum_{i=1}^{9}(3d_i^2-2)y_i \quad (47)$$

$$\hat{b} = \frac{1}{6}\sum_{i=1}^{9} d_i y_i \quad (48)$$

$$\hat{c} = \Delta^{-1}\sum_{i=1}^{9}(2\beta-2d_i^2)y_i \quad (49)$$

The expression of the distance of the axis of the paraboloid (i.e. of the estimate of the edge) from the centre of the 3x3 neighbourhood is then:

$$\delta = -\frac{3\beta-2}{2}\frac{\sum_{i=1}^{9} d_i y_i}{\sum_{i=1}^{9}(3d_i^2-2)y_i} \quad (50)$$

The expression for the normalized error of fit is:

$$\epsilon_{i,j} = \frac{\|\mathbf{y}-\hat{\mathbf{y}}\|}{\|\mathbf{y}\|} = 1 - \frac{\|\hat{\mathbf{y}}\|}{\|\mathbf{y}\|} = 1 - \frac{\|A\hat{\mathbf{a}}\|}{\|\mathbf{y}\|} \quad (51)$$

Acknowledgements

The derivation of Rice's formula in the appendix was done in collaboration with Massimo Porrati. P.P. conducted part of this research while (a) visiting CEREMADE at the Université Paris-Dauphine, (b) with the Electrical Engineering and Computer Science Department at U.C.Berkeley, (c) with the International Computer Science Institute in Berkeley. This research was partially supported by NSF Presidential Young Investigator Award IRI-8957274 to J.M. and by U.S. Army Research Office grant number DAAL 03-86-K-0171 (P.P.). The experiments reported in this paper have been conducted using Paul Kube's 'viz' image processing package. We are very grateful to Paul for providing us with the code and with solicitous assistance in its use.

References

[1] Edward Adelson and James Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America*, 2(2):284–299, 1985.

[2] David J. Beymer. *Junctions: their detection and use for grouping in images.* Master's thesis, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1989.

[3] T.O. Binford. Inferring surfaces from images. *Artificial Intelligence*, 17:205–244, 1981.

[4] Andrew Blake and Andrew Zisserman. *Visual reconstruction.* MIT press, 1987.

[5] John Canny. A computational approach to edge detection. *IEEE trans. PAMI*, 8:679–698, 1986.

[6] D. Forsyth and A. Zisserman. Mutual illumination. In *Proceedings of the IEEE CVPR*, pages 466–473, 1989.

[7] William Freeman and Edward Adelson. Steerable filters for image analysis. Technical Report 126, MIT, Media Laboratory, 1990.

[8] Goesta H. Granlund. In search of a general picture processing operator. *Computer Graphics and Image Processing*, 8:155–173, 1978.

[9] B.K.P. Horn. Image intensity understanding. *Artificial intelligence*, 8(2):201–231, 1977.

[10] Jitendra Malik and Pietro Perona. A computational model of texture segmentation. In *IEEE Computer Society Computer Vision and Pattern Recognition conference proceeding*, pages 326–332, San Diego, June 1989.

[11] M.C.Morrone, D.C. Burr, J. Ross, and R. Owens. Mach bands depend on spatial phase. *Nature*, (324):250–253, 1986.

[12] M.C. Morrone and D.C. Burr. Feature detection in human vision: a phase dependent energy model. *Proc. R. Soc. Lond. B*, 235:221–245, 1988.

[13] M.C. Morrone and R.A. Owens. Feature detection from local energy. *Pattern Recognition Letters*, 6:303–313, 1987.

[14] R. Owens, S. Venkatesh, and J. Ross. Edge detection is a projection. *Pattern Recognition Letters*, 9:233–244, 1989.

[15] Pietro Perona. Finite representation of deformable functions. Technical Report 90-034, International Computer Science Institute, 1947 Center st., Berkeley CA 94704, 1990.

[16] J. Ponce and M. Brady. Towards a surface primal sketch. Technical Report 824, MIT Artificial Intelligence Laboratory, 1985.

[17] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1988.

[18] S.O.Rice. Mathematical analysis of random noise. *Bell System Technical Journal*, 24:46–156, 1945.

A Some useful formulae

A.1 derivatives of $W(x)$

$$\begin{aligned} W(x) &= (I * \mathbf{f}^T)(\mathbf{f} * I) = \|I * \mathbf{f}\|^2 && (52)\\ \frac{1}{2}W'(x) &= (I * \mathbf{f}^T)(\mathbf{f}' * I) && (53)\\ \frac{1}{2}W''(x) &= (I * \mathbf{f}^T)(\mathbf{f}'' * I) + (I * \mathbf{f}'^T)(\mathbf{f}' * I) && (54)\\ &= (I * \mathbf{f}^T)(\mathbf{f}'' * I) + \|I * \mathbf{f}'\|^2 && (55) \end{aligned}$$

If $I = G = c_1\delta^{(-2)} + c_2\delta^{(-1)} + c_3\delta$ then:

$$I * \mathbf{f} = \sum_{j=1\ldots3} c_j \mathbf{f}^{(j-3)} = \sum_{j=1\ldots3} c_j \mathbf{F}^{(j-1)} = \mathbf{Hc} \qquad (56)$$

where $\mathbf{c}^T = [c_1, c_2, c_3]$ and $\mathbf{H}$ is defined componentwise by $\mathbf{H}_{ij} = F_i^{(j-1)}$, $j = 1, \ldots, 3$ and $i = 1, \ldots, n$. In this case the expressions for $W(x)$ and its derivatives become:

$$\begin{aligned} W_{G_c}(x) &= \mathbf{c}^T\mathbf{H}^T\mathbf{Hc} = \|\mathbf{Hc}\|^2 && (57)\\ \frac{1}{2}W'_{G_c}(x) &= \mathbf{c}^T\mathbf{H}'^T\mathbf{Hc} && (58)\\ \frac{1}{2}W''_{G_c}(x) &= \mathbf{c}^T\mathbf{H}''^T\mathbf{Hc} + \mathbf{c}^T\mathbf{H}'^T\mathbf{H}'\mathbf{c} && (59)\\ &= \mathbf{c}^T\mathbf{H}''^T\mathbf{Hc} + \|\mathbf{H}'\mathbf{c}\|^2 && (60) \end{aligned}$$

where $\mathbf{H}'$, and $\mathbf{H}''$ are obtained differentiating $\mathbf{H}$ componentwise.

B Distance between maxima

In this section we compute the average spacing between positive-derivative zero crossings of filtered white noise. We summarize the result in the following lemma and subsequently proceed with the derivation.

Theorem. Consider the signal r(x) defined by:

$$r(x) = \sum_i \alpha_i(x)(f_i * \eta)(x) = \vec{\alpha}(x)^T(\mathbf{f} * \eta)(x) \qquad (61)$$

where η is a white gaussian process. Then the average spacing between the positive-derivative zero crossings of r(x) is:

$$T = 2\pi\frac{\sqrt{a(x)c(x) - b^2(x)}}{c(x) - b(x)} \qquad (62)$$

With a, b, abd c defined by

$$\begin{aligned} a(x) &= \vec{\alpha}'(x)^T\mathbf{R}_f\vec{\alpha}'(x) + 2\vec{\alpha}'(x)^T\mathbf{R}_{ff'}\vec{\alpha}(x) + \vec{\alpha}(x)^T\mathbf{R}_{f'}\vec{\alpha}(x)\\ b(x) &= \vec{\alpha}'(x)^T\mathbf{R}_f\vec{\alpha}''(x) + 2\vec{\alpha}'(x)^T\mathbf{R}_{ff'}\vec{\alpha}'(x) + \vec{\alpha}'(x)^T\mathbf{R}_{ff''}\vec{\alpha}(x)\\ &\quad \vec{\alpha}(x)^T\mathbf{R}_{f'f}\vec{\alpha}''(x) + 2\vec{\alpha}(x)^T\mathbf{R}_{f'}\vec{\alpha}'(x) + \vec{\alpha}(x)^T\mathbf{R}_{f'f''}\vec{\alpha}(x)\\ c(x) &= \vec{\alpha}''(x)^T\mathbf{R}_f\vec{\alpha}'' + 4\vec{\alpha}'(x)^T\mathbf{R}_{f'}\vec{\alpha}'(x) + \vec{\alpha}(x)^T\mathbf{R}_{f''}\vec{\alpha}(x)\\ &\quad +4\vec{\alpha}''(x)^T\mathbf{R}_{ff'}\vec{\alpha}'(x) + 2\vec{\alpha}''(x)^T\mathbf{R}_{ff''}\vec{\alpha}(x) + 2\vec{\alpha}'(x)^T\mathbf{R}_{f'f''}\vec{\alpha}(x) \end{aligned} \qquad (63)$$

where the $\mathbf{R}$'s are the correlation matrices of the functions f_i, f_i', f_i''.

Corollary. (Rice) Consider the signal r(x) defined by:

$$r(x) = (f * \eta)(x) \tag{64}$$

where η is a white gaussian process. Then the average spacing between the positive-derivative zero crossings of r(x) is:

$$T_0 = \frac{1}{Es} = 2\pi\sqrt{\frac{R_f(0)}{R_f''(0)}} \tag{65}$$

where the **R**'s are the autocorrelation matrices of the functions f and f'.

B.1 The shift-invariant case

We first rederive the result due to Rice [18]: the average spacing T between positive-derivative zero crossings of filtered Gaussian white noise. Consider a differentiable convolution kernel $f(x)$ and consider the signal $r(x)$ defined by:

$$r(x) \doteq (f * \eta)(x) \tag{66}$$

Necessary and sufficient condition for x_i to be a positive-derivative zero of r is that $r(x_i) = 0$ and $r'(x_i) > 0$. Consider now the new signal $s(x)$ defined by:

$$s(x) \doteq -\delta(r(x))r'(x)H(r'(x)) \tag{67}$$

where δ and H indicate respectively the delta distribution and the Heavyside step function. Notice that $s(x)$ is the sum of unit-volume delta functions located at the positions of the positive-derivative zeros of $r(x)$ (see also Fig. 8). Then the spatial average of $r(x)$ is the reciprocal of the quantity we want to compute. Because of the ergodicity properties of $s(x)$ we can compute this spatial average as an expectation at any point x:

$$\frac{1}{T} = Es(x) \tag{68}$$

Now remember that:

$$\delta(x) = \int_{\mathbb{R}} e^{i2\pi px}dp \tag{69}$$

$$H(x) = \int_{\mathbb{R}} \frac{-i}{2\pi q}e^{i2\pi qx}dq \tag{70}$$

$$xg(x) = \int_{\mathbb{R}} \hat{g}(q)\frac{1}{i2\pi}\frac{\partial}{\partial q}e^{i2\pi qx}dq \tag{71}$$

So we can write:

$$Es = \int_{\mathbb{R}^2} \frac{-i}{2\pi}\frac{-i}{2\pi q}\frac{\partial}{\partial q}Ee^{i2\pi(pr+qr')}dpdq \tag{72}$$

The expectation of the exponential is (call $k = i2\pi$ for convenience)

$$Ee^{k(pr+qr')} = \int_{\mathcal{N}} e^{k(pf*\eta+qf'*\eta)}P_{\mathrm{H}}(\eta)d\eta \tag{73}$$

Where $\mathcal{N}$ is the sample space of the random process η, and $P_{\mathrm{H}}(\eta) = e^{-\frac{1}{2}\int_{\mathbb{R}}\eta^2(\nu)d\nu}$ is the probability density. Expanding the convolution integrals and the probability density, and making use of the fact that the correlation function of a function and its derivative is zero in the origin, we obtain:

$$Ee^{k(pr+qr')} = \int_{\mathcal{N}} e^{k(p\int_{\mathbb{R}} f(x-\mu)\eta(\mu)d\mu+q\int_{\mathbb{R}} f'(x-\lambda)\eta(\lambda)d\lambda)}e^{-\frac{1}{2}\int_{\mathbb{R}}\eta^2(\nu)d\nu}d\eta$$

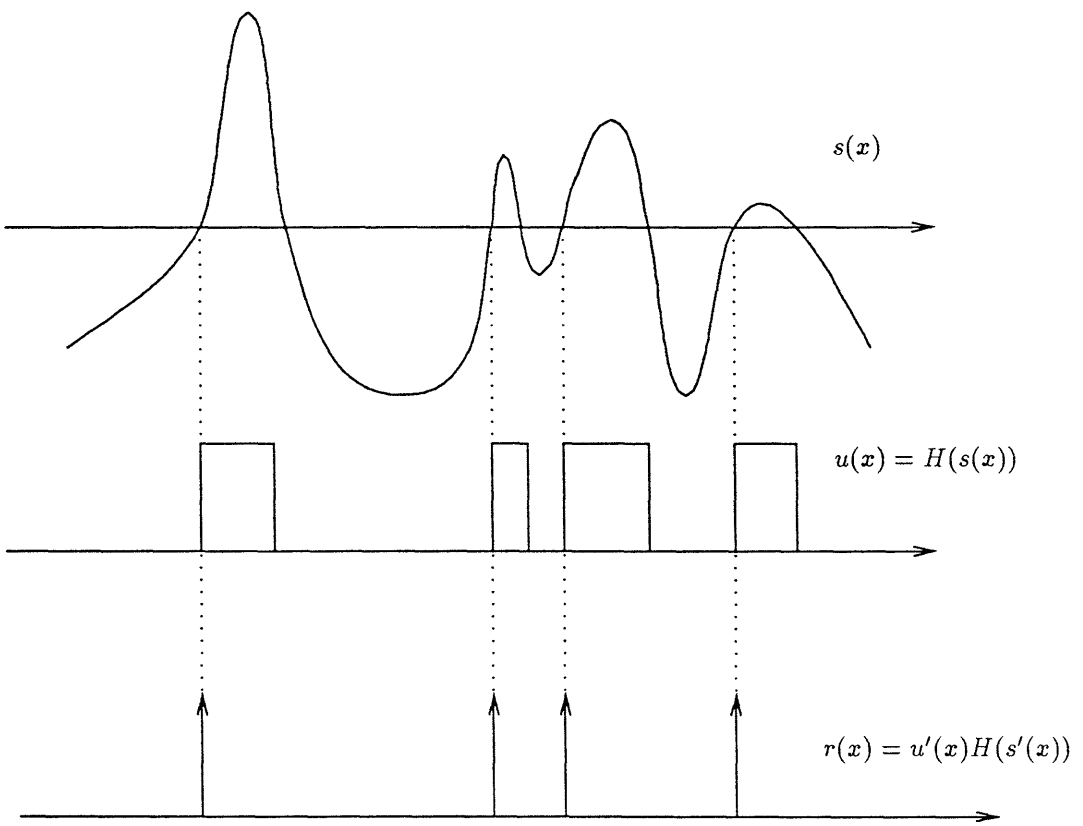

Figure 8: Trick for computing the density of the zeros of a function $s(x)$. Details in the text or elsewhere.

$$
\begin{aligned}
&= \int_{\mathcal{N}} e^{-\frac{1}{2}\int_{\mathbb{R}} \eta^2(\mu) - 2k[pf(x-\mu)+qf'(x-\mu)]\eta(\mu)d\mu} d\eta \\
&= \int_{\mathcal{N}} e^{-\frac{1}{2}\int_{\mathbb{R}} (\eta(\mu)-k[pf(x-\mu)+qf'(x-\mu)])^2 - k^2(pf(x-\mu)+qf'(x-\mu))^2 d\mu} d\eta \\
&= e^{\frac{1}{2}k^2\int_{\mathbb{R}}(pf(x-\mu)+qf'(x-\mu))^2 d\mu} \\
&= e^{\frac{1}{2}k^2\int_{\mathbb{R}}(pf(\mu)+qf'(\mu))^2 d\mu} \\
&= e^{\frac{1}{2}k^2(p^2R_f(0)+q^2R_f''(0))} \qquad (74)
\end{aligned}
$$

Substituting into (72) we obtain:

$$
\begin{aligned}
Es &= \int_{\mathbb{R}^2} \frac{1}{q}\frac{-1}{(2\pi)^2}\frac{\partial}{\partial q} e^{-\frac{1}{2}(2\pi)^2(p^2R_f(0)+q^2R_f''(0))} dpdq \\
&= \int_{\mathbb{R}^2} R_f''(0) e^{-\frac{1}{2}(2\pi)^2(p^2R_f(0)+q^2R_f''(0))} dpdq \qquad (75) \\
&= \frac{2\pi R_f''(0)}{\sqrt{(2\pi)^4 R_f(0)R_f''(0)}} \\
&= \frac{1}{2\pi}\sqrt{\frac{R_f''(0)}{R_f(0)}} \qquad (76)
\end{aligned}
$$

So our estimate becomes:

$$
T_0 = \frac{1}{Es} = 2\pi\sqrt{\frac{R_f(0)}{R_f''(0)}} \qquad (77)
$$

which is Rice's formula.

This may be generalized to positive-derivative crossings of any level L by considering the zeros of $r(x)-L$. The same computations carry through generating a linear term pL in the exponent of Eq. (75). This requires a shift in the p coordinates to solve the integral generating an additional factor in the solution:

$$
T_L = \frac{1}{Es} = 2\pi\sqrt{\frac{R_f(0)}{R_f''(0)}} e^{\frac{L}{2\pi R_f^{1/2}}} \qquad (78)
$$

We may use this formula for computing the average distance T_M between maxima of the signal r : the maxima are the positive-derivative zeros of $-r'$:

$$
T_M = \frac{1}{Es} = 2\pi\sqrt{\frac{R_{f'}(0)}{R_{f'}''(0)}} \qquad (79)
$$

As an upper bound estimate of the average distance between maxima above a threshold L we may use the average distance between positive-derivative L level crossings (78).

B.2 The shift-variant case

In this section the more general case where

$$r(x) = \sum_i \alpha_i(x)(f_i * \eta)(x) = \vec{\alpha}(x)^T(\mathbf{f} * \eta)(x) \tag{80}$$

is considered.

We first compute the derivatives of r, they will be useful later:

$$\begin{aligned} r'(x) &= \vec{\alpha}'(x)^T(\mathbf{f} * \eta)(x) + \vec{\alpha}(x)^T(\mathbf{f}' * \eta)(x) \\ r''(x) &= \vec{\alpha}''(x)^T(\mathbf{f} * \eta)(x) + 2\vec{\alpha}'(x)^T(\mathbf{f}' * \eta)(x) + \vec{\alpha}(x)^T(\mathbf{f}'' * \eta)(x) \end{aligned} \tag{81}$$

Since we want to compute the inter-positive–maxima distance we substitute r' to r in equation (67) and the following. Therefore we need to compute $Ee^{k(pr'+qr'')}$ which may be done along the lines of the calculations in (74). The convolution kernels are different therefore the integral at the exponent of the exponential in the last lines of (74) becomes:

$$\begin{aligned} &\int_{\mathbb{R}} (p(\vec{\alpha}'(x)^T\mathbf{f}(x-\mu) + \vec{\alpha}(x)^T\mathbf{f}'(x-\mu)) + \\ &q(\vec{\alpha}''(x)^T\mathbf{f}(x-\mu) + 2\vec{\alpha}'(x)^T\mathbf{f}'(x-\mu) + \vec{\alpha}(x)^T(\mathbf{f}''(x-\mu))))^2 d\mu \end{aligned} \tag{82}$$

Expanding the square one gets

$$p^2 \int_{\mathbb{R}} A(x,\mu)d\mu + 2pq \int_{\mathbb{R}} B(x,\mu)d\mu + q^2 \int_{\mathbb{R}} C(x,\mu)d\mu \tag{83}$$

Where the functions A, B and C are the sums:

$$\begin{aligned} A(x,\mu) &= \vec{\alpha}'(x)^T\mathbf{f}(x-\mu)\mathbf{f}^T(x-\mu)\vec{\alpha}' + 2\vec{\alpha}'(x)^T\mathbf{f}(x-\mu)\mathbf{f}'^T(x-\mu)\vec{\alpha}(x) \\ B(x,\mu) &= \vec{\alpha}'(x)^T\mathbf{f}(x-\mu)\mathbf{f}^T(x-\mu)\vec{\alpha}''(x) + \dots \\ C(x,\mu) &= \vec{\alpha}''(x)^T\mathbf{f}(x-\mu)\mathbf{f}^T(x-\mu)\vec{\alpha}''(x) + \dots \end{aligned} \tag{84}$$

Denote with the symbol $\mathbf{R}$ the correlation matrices:

$$\begin{aligned} (\mathbf{R}_f)_{ij} &= \int_{\mathbb{R}} f_i(\mu)f_j(\mu)d\mu \\ (\mathbf{R}_{f'})_{ij} &= \int_{\mathbb{R}} f_i'(\mu)f_j'(\mu)d\mu \\ (\mathbf{R}_{ff'})_{ij} &= \int_{\mathbb{R}} f_i(\mu)f_j'(\mu)d\mu \\ (\mathbf{R}_{f'f})_{ij} &= \int_{\mathbb{R}} f_i'(\mu)f_j(\mu)d\mu \\ &\dots \end{aligned}$$

Then the integrals of A, B and C, $a(x) = \int A(x,\mu)d\mu$ etc. , become

$$\begin{aligned} a(x) &= \vec{\alpha}'(x)^T\mathbf{R}_f\vec{\alpha}'(x) + 2\vec{\alpha}'(x)^T\mathbf{R}_{ff'}\vec{\alpha}(x) + \vec{\alpha}(x)^T\mathbf{R}_{f'}\vec{\alpha}(x) \\ b(x) &= \vec{\alpha}'(x)^T\mathbf{R}_f\vec{\alpha}''(x) + 2\vec{\alpha}'(x)^T\mathbf{R}_{ff'}\vec{\alpha}'(x) + \vec{\alpha}'(x)^T\mathbf{R}_{ff''}\vec{\alpha}(x) \\ &\quad \vec{\alpha}(x)^T\mathbf{R}_{f'f}\vec{\alpha}''(x) + 2\vec{\alpha}(x)^T\mathbf{R}_{f'}\vec{\alpha}'(x) + \vec{\alpha}(x)^T\mathbf{R}_{f'f''}\vec{\alpha}(x) \\ c(x) &= \vec{\alpha}''(x)^T\mathbf{R}_f\vec{\alpha}'' + 4\vec{\alpha}'(x)^T\mathbf{R}_{f'}\vec{\alpha}'(x) + \vec{\alpha}(x)^T\mathbf{R}_{f''}\vec{\alpha}(x) \\ &\quad +4\vec{\alpha}''(x)^T\mathbf{R}_{ff'}\vec{\alpha}'(x) + 2\vec{\alpha}''(x)^T\mathbf{R}_{ff''}\vec{\alpha}(x) + 2\vec{\alpha}'(x)^T\mathbf{R}_{f'f''}\vec{\alpha}(x) \end{aligned} \tag{85}$$

Therefore we have (cfr. equation (75)):

$$
\begin{aligned}
Es &= \int_{\mathbb{R}^2} \frac{1}{q} \frac{-1}{(2\pi)^2} \frac{\partial}{\partial q} e^{-\frac{1}{2}(2\pi)^2(p^2a(x)+q^2c(x)+2pqb(x)} dpdq \\
&= \int_{\mathbb{R}^2} c(x) e^{-\frac{1}{2}(2\pi)^2(p^2a(x)+q^2c(x)+2pqb(x))} dpdq + \\
&\quad + \int_{\mathbb{R}^2} \frac{p}{q} b(x) e^{-\frac{1}{2}(2\pi)^2(p^2a(x)+q^2c(x)+2pqb(x))} dpdq \\
&= \frac{1}{2\pi} \frac{c(x) - b(x)}{\sqrt{a(x)c(x) - b^2(x)}} \qquad (86)
\end{aligned}
$$

So that the spacing between positive maxima of $r(x)$ is:

$$
T = \frac{1}{Es} = 2\pi \frac{\sqrt{a(x)c(x) - b^2(x)}}{c(x) - b(x)} \qquad (87)
$$

With a, b, abd c defined above. Notice that when α is a constant (i.e. $\alpha' = 0$, $\alpha'' = 0$) equation (87) reduces to equation (79).