Analytical and Empirical Performance Evaluation of Subpixel Line and Edge Detection

Carsten Steger
Forschungsgruppe Bildverstehen (FG BV), Informatik IX
Technische Universität München, Orleansstr. 34, 81667 München
Phone: +49 (89) 48095-211, Fax: +49 (89) 48095-203
E-mail: stegerc@informatik.tu-muenchen.de

Abstract

An extensive analysis of the quality of the extraction results of a subpixel line detector and a subpixel edge detector is carried out. The localization quality of line and edge points in the presence of noise is analyzed analytically, and new formulas describing this relationship are derived. Tests on synthetic noisy images show the formulas to hold very well in practice. Experiments on real images show that subpixel accuracy better than one tenth of a pixel is possible in typical industrial inspection tasks.

1: Introduction

The analysis of the performance characteristics of a feature extraction algorithm is very important [8]. First, it makes an algorithm comparable to other algorithms, thus helping users in selecting the appropriate method for the task they have to solve. Second, it helps to identify breakdown points of the algorithm, i.e., areas where the algorithm cannot be used because some of the assumptions it makes are violated. Therefore, in this paper an attempt is made to characterize the performance of the subpixel line and edge detection algorithms proposed in [15, 16, 17]. The study is mainly concerned with the subpixel precision and accuracy of the extracted line and edge positions and line widths because of their importance for many applications, especially industrial inspection. Following [8], for the purpose of this paper precision refers to the repeatability of the extraction results, i.e., their variance, and accuracy refers to their absolute errors, including possible bias.

2: Subpixel line and edge detection algorithms

The line detector to be analyzed is discussed in detail in [15, 16, 17]. Therefore, only a brief overview is given here. The algorithm models lines as curves $s(t)$ that exhibit a characteristic 1D line profile in the direction perpendicular to the line. The most relevant type of line profile for applications is the asymmetrical bar-shaped profile given by

$$f_a(x) = \begin{cases} 0, & x < -w \\ 1, & |x| \le w \\ a, & x > w \ , \end{cases} \tag{1}$$

where w is half the line width and $a \in [0, 1]$ is the asymmetry. General lines of contrast h can be obtained by considering a scaled asymmetrical profile.

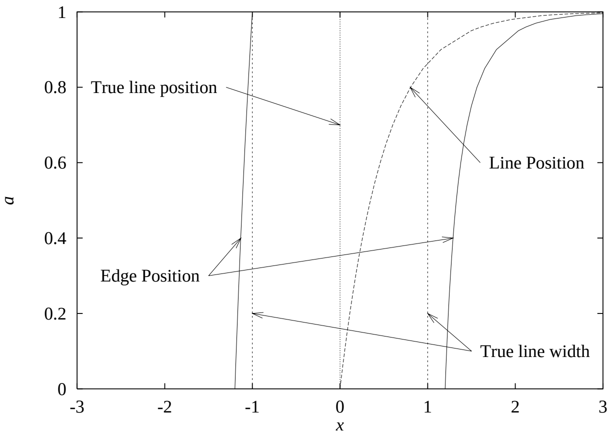


Figure 1. Position of an asymmetrical line and its corresponding edges with width $w = 1$, $\sigma = 1$, and $a \in [0, 1]$.

In 1D, line positions can be extracted by convolving $f_a(x)$ with the derivatives of a Gaussian smoothing kernel $g_\sigma(x)$ of width σ. This leads to a scale-space description of the model line profile:

$$\begin{aligned} r_a(x, \sigma, w, a) &= \phi_\sigma(x + w) + (a - 1)\phi_\sigma(x - w) & (2) \\ r'_a(x, \sigma, w, a) &= g_\sigma(x + w) + (a - 1)g_\sigma(x - w) & (3) \\ r''_a(x, \sigma, w, a) &= g'_\sigma(x + w) + (a - 1)g'_\sigma(x - w)\ . & (4) \end{aligned}$$

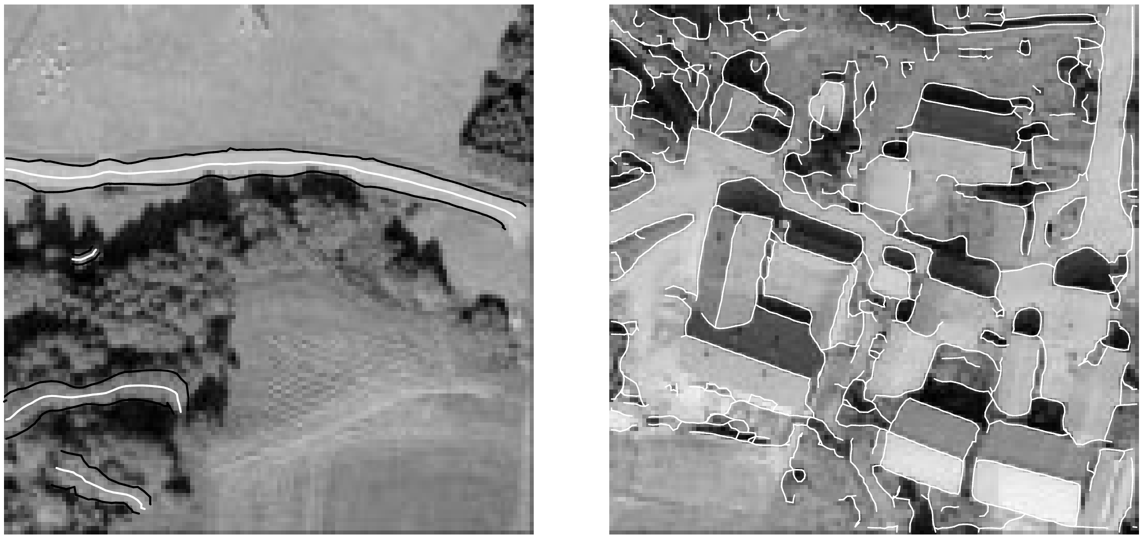
Line points are given by the points where $r'_a(x, \sigma, w, a) = 0$. Salient lines are selected based on the magnitude of the second derivative $r''_a(x, \sigma, w, a)$. For the extraction of the line width, the two edges to the left and right of the line position need to be extracted. They are given by the points where $r''_a(x, \sigma, w, s) = 0$ and $r'''_a(x, \sigma, w, a)r'_a(x, \sigma, w, a) < 0$. It can be shown that Gaussian smoothing necessarily leads to a bias of the extracted line positions if $a \neq 0$. The biased line position is given by

$$l = -\frac{\sigma^2}{2w} \ln(1 - a)\ . \qquad (5)$$

Similarly, the edge position will generally be biased. An example of this behavior for $w = 1$, $\sigma = 1$, and $a \in [0, 1]$ is shown in Figure 1. It can be seen that as a gets larger the line and edge positions are pushed to the weak side, i.e., the side that possesses the smaller edge gradient. Furthermore, the line width is extracted too large for all a.

In [15, 16, 17] it is shown that the bias of the line position and widths can be predicted analytically. The true values of the line width and asymmetry are mapped to the line width and gradient ratio, which are extractable from the image. Since this map is invertible, the bias can be removed by plugging the extracted features into the inverted bias prediction.

In 2D, lines are modeled as curves $s(t)$ that exhibit the characteristic profile f_a in the direction perpendicular to the line, i.e., perpendicular to $s'(t) = n(t)$. This means that the first directional derivative in the direction $n(t)$ must vanish, while the second directional derivative should be of large absolute value. The necessary 1D zero crossing is obtained with subpixel accuracy by extrapolating it from a local second-order Taylor polynomial. The direction $n(t)$ is obtained from the eigenvector corresponding to the largest eigenvalue of the Hessian matrix of the image convolved with a Gaussian. To detect the width of the line, for each line point the closest points to the left



(a) Extracted lines

(b) Extracted edges

Figure 2. Lines and their width extracted in an aerial image (a). Edges extracted in an aerial image (b). Lines and edges are displayed in white, while the line width is displayed in black.

and right of the line point, i.e., along $-n$ and n, where the absolute value of the gradient takes on its maximum value, are determined. This is done by regarding the edges of the line as lines in the gradient image, and using the same techniques as mentioned above to achieve subpixel accuracy. The extracted line width and gradient ratio are plugged into the inverted bias function to achieve unbiased extraction results. The unbiasedness of the extracted line position, width, contrast, asymmetry, and orientation was established for noise-free synthetic images in [17]. Figure 2(a) shows an example of the lines extracted with this approach. Note that the lines are unbiased in the entire image, especially in the highly asymmetrical part in the center of the road.

The above mentioned approach to regard edges as lines in the gradient image leads quite naturally to a subpixel precise edge detector [17]. An example of the edges extracted with this approach is shown in Figure 2(b).

3: Quality of the localization of line and edge points

For real images, noise plays a significant role. Different “types” of noise can be distinguished: The first kind is characterized by random fluctuations of the image intensity. This is the “classical” notion of random noise. A second kind of “noise” is what might be called “clutter noise” or “structured noise,” i.e., random structures that have the same appearance as the objects of interest. In this paper, only random noise in the classical sense will be considered for the following reasons: First, in the intended application area of industrial inspection one usually has excellent control over the lighting conditions. Often, objects are back-lit, and hence clutter noise can be eliminated à priori. Furthermore, the distinction between clutter noise and salient objects often involves a distinction on a semantical level, which cannot be decided in a low-level feature extractor. For example, in road extraction applications from low-resolution images roads can be detected as lines. However, house roofs often appear as short lines, and thus can be considered as clutter noise *in this particular application*, although for the low-level feature extractor they are perfectly good lines. Therefore, a meaningful performance evaluation of an algorithm in the presence of clutter noise seems only possible using the semantics of a particular application. For example, an evaluation of the line extractor discussed here and in [15, 16, 17] for the purpose of road extraction can be found in [10].

Random noise plays a significant role for the precision and accuracy of the extracted features, and therefore is of major importance for industrial inspection, where objects often have to be measured with subpixel precision and accuracy because of the limited resulution of the imaging sensor. This kind of noise enters the image in different stages of the imaging process, e.g., through the photon flux on the sensor or the quantization of the image intensity to a finite number of gray levels. The noise $n(x)$ can be modeled as a random process in one dimension and as a random field in higher dimensions. Thus, the observed version of the image is given by

$$i(x,y) = f(x,y) + n(x,y) \ . \tag{6}$$

The noise component n is assumed to be wide-sense stationary, statistically independent of the image content f, and white, i.e., its expected value $E(n) = 0$ and its autocorrelation $R(\tau) = \sigma_n^2 \delta(\tau)$, where σ_n^2 is the variance of the noise [13].

Obviously, if noise is added to an image the extracted line or edge positions, and in general any extraction results, such as the line widths, will not be the correct values that would have been obtained from the uncorrupted version of the image. Therefore, one of the major performance criteria of an algorithm is the dependence of the extracted features on the input noise level. Put into different terms, what needs to be determined is how the variance of the line position and width and the variance of the edge position varies as the variance of the noise in the image varies. Ideally, if this dependency has been determined the algorithm could return an estimate of the variance of the respective feature, e.g., the line position, if given an estimate of the variance of the image noise. The variance σ_n of the image noise can, for example, be determined by automatic procedures from an input image [2, 7]. Such a variance estimate can serve as an aid for self-diagnosis [8], e.g., to determine if certain requirements of the output data can be met. For example, if an application requires one tenth of a pixel accuracy in the line or edge position, and the returned standard deviation of the line position is larger than some fraction of this amount, an algorithm can alert the user to the fact that the lines cannot be extracted with the required precision. The user could then try to fix the problem, e.g., by changing the lighting conditions or by using a different sensor to reduce the noise level in the images.

3.1: Previous work on localization quality

Over the years, several attempts have been made to characterize the quality of the extracted feature positions. Most of them were made for edge detectors. Often, these quality measures are made in connection with deriving an "optimal" operator with respect to several criteria, one of which often is a term that tries to measure the goodness of edge locations. One of the earliest approaches to this problem is introduced in [3], where an optimal 1D edge detector with respect to three criteria is derived: good detection, i.e., the algorithm should have a high probability of extracting true edge points, while having a low probability of extracting false edge points, good localization, i.e., the extracted edge points should be as close as possible to the true edge points, and finally having only a single response to an edge. The optimal operator is described by a convolution of the image with a kernel of finite width $2w$. It is assumed to yield a maximum response at the edge location, i.e., the first derivative of the filter response at the edge is zero. With this, the localization criterion can be derived as follows: Assume the response of convolving the image i with the edge operator e to be given by

$$r(x) = h(x) * i(x) = h(x) * f(x) + h(x) * n(x) = r_f(x) + r_n(x) \ . \tag{7}$$

Then, due to the image noise n, the extracted edge location will lie at a point x_e, where $r'(x_e) = 0$. Thus, $r'_n(x_e) + r'_f(x_e) = 0$. If this equation is expanded into a Taylor series of order 1 about the origin, one obtains $r'(x_e) = r'(0) + r''(0)x_e + O(x_e^2)$. Since the model edge is located at 0, by assumption $r'(0) = 0$, and hence $x_e \approx -r'_n(x_e)/r''(0)$. In [3] it is now claimed that $r'_n(x_e)$ is a Gaussian random variable with variance

$$E(r'_n(x_e)^2) = \sigma_n^2 \int_{-w}^{w} f'(x)^2 \, dx \ . \tag{8}$$

Thus, the variance of the edge position is given by

$$E(x_e^2) = \frac{\sigma_n^2 \int\limits_{-w}^{w} f'(x)^2 \, dx}{\left(\int\limits_{-w}^{w} h'(-x) f'(x) \, dx \right)^2} \ . \tag{9}$$

These continuous criteria for an optimal edge detector are adapted for pixel-precise discrete edge operators in [4], where mainly the formulation of the single response criterion changes.

This definition of the localization quality measure was later criticized for being incorrect [19]. The major objection put forth there is that (8) only holds if x_e is constant for all realizations, which, of course, is not the case. Another objection is that the localization criterion in [3] only takes into account the first zero crossing of $r'(x)$, whereas it should take into account all zero crossings of the response. Therefore, the physical density of the zero crossings, i.e., the expected number of zero crossings per unit time, of the edge detector applied to a noisy step edge is derived in [19]. This density is given by

$$\mu(x) = \frac{1}{2\pi} \sqrt{-\frac{R''_{n'}(0)}{R_{n'}(0)}} \exp(-f(x)^2/2\sigma_{n'}^2) \ . \tag{10}$$

The authors note that $1 - \mu(x)$ is a measure for the suppression of the density of zero crossings by the edge operator, and go on to prove that the first derivative of the Gaussian kernel is the unique function that optimizes this criterion. With this criterion, in effect the localization and single response criteria used in [3] are combined into a single criterion, yielding a better optimality criterion for edge operators. Of course, $\mu(x)$ is not very useful to characterize the dependency of the localization of edges on the image noise, precisely for the fact that all zero crossings are taken into account.

For this reason, an attempt is made in [12] to derive the true variance of the edge position. The authors note that in addition to the problems mentioned in [19], i.e., the wrong derivation of $E(r'(x)^2)$, the Taylor expansion of the edge location should be done differently. They claim that the correct Taylor expansion is given by $r'(x_e) = r'_n(0) + (r''_f(0) + r''_n(0))x_e$, and hence $x_e \approx -r'_n(0)/(r''_f(0) + r''_n(0))$. They then go on to derive the probability density of x_e and arrive at the result that it is given by a sum of two Cauchy densities. Therefore, they note, the variance of x_e does not exist. Of course, in this case the mean of x_e also would not exist. What this obviously erroneous result implies is that if even the tiniest amount of noise is added to an image, the edge locations would fluctuate wildly all over the image, which is, of course, not supported by experimental evidence. If this were true, edges could in effect not be extracted at all because their positions would be meaningless.

A different approach is given in [1], where the imaging process is modeled in several steps. First, the ideal continuous image is assumed to be blurred by the imaging device with a Gaussian kernel with standard deviation σ_b. Then, the image is sampled and noise with standard deviation σ_n is added to the image. Finally, edge extraction is done by smoothing the image with a Gaussian kernel with standard deviation σ_e and calculating the gradient of the resulting image. The authors give the following formula for the variance of an edge of height h, if the edge extraction is performed in a direction forming an angle α with the direction perpendicular to the edge:

$$E(x_e^2) = \frac{3\sigma_n^2(\sigma_b^2 + \sigma_e^2)^3}{8h^2\sigma_e^6 \cos^6 \alpha} \quad . \tag{11}$$

Unfortunately, almost no details of the derivation of (11) are given, and hence it is hard to judge whether (11) is correct.

Contrary to this, edge detection is regarded as template matching in [8]. In this case, the variance of the edge position is given by

$$\sigma_e^2 = \frac{\sigma_n^2}{\sum\limits_{r=-w}^{w} \sum\limits_{c=-w}^{w} f_u^2(r,c)} \quad , \tag{12}$$

where the template is of size $(2w + 1) \times (2w + 1)$ and $f_u(r,c)$ is the derivative of the template in the direction perpendicular to the edge. Unfortunately, since the edge and line detection algorithms discussed in Section 2 do not use template matching, this result does not apply to them.

3.2: Variance of the zero crossing of an arbitrary function

Because none of the above derivations of the variance of the edge position are applicable to the problem of determining the variance of the line position and width, and because some of them are erroneous even for the problem of determining the variance of the edge position, there is a need for an, at least qualitatively, correct formula that also is supported by experimental evidence. The algorithm of determining the line position in the 1D continuous case can be broken down into two steps: the input signal is first convolved with the derivative of a Gaussian kernel, and then the zero crossings of the resulting response function $r'(x)$ are determined, where additionally $r''(x) < 0$ is required. For edges, the only change is that the zero crossings of the second derivative are determined. Since convolution with a Gaussian kernel is a linear operation, the response $r(x)$ of the line detection operator to the corrupted signal $i(x)$ is given by

$$r(x) = g_\sigma(x) * i(x) = g_\sigma(x) * f(x) + g_\sigma(x) * n(x) = r_f(x) + r_n(x) \tag{13}$$

and analogously for the corresponding derivatives. Thus, the response of the line detection operator to a model line profile $f(x)$ corrupted by wide-sense stationary white noise $n(x)$ can be broken down into a deterministic part $r_f(x)$ and a stochastic process $r_n(x)$. The deterministic part can, of course, be determined analytically, e.g., by (2)–(4) for the asymmetrical bar-shaped profile. Note that the deterministic part of the response has exactly one zero crossing of $r'(x)$ in $[-\infty, \infty]$. The random part of the response shifts this zero crossing and may introduce additional zero crossings of $r'(x)$. In order to determine the localization quality of the line detection algorithm, it is therefore useful to consider the variance of the zero crossing of $r'(x)$ closest to the zero crossing of the deterministic part $r'_f(x)$ of the response.

Since the noise $n(x)$ is assumed to be wide-sense stationary and white, the statistics of the response of the line detector to the noise can be calculated easily [13, 18]. Because the mean value $E(n(x))$ is assumed to be zero, convolution of $n(x)$ with an arbitrary derivative of the Gaussian kernel is also a stationary stochastic process with zero mean. Furthermore, the autocorrelation function $R_r(\tau)$ of the noise smoothed with a Gaussian kernel is given by [13]

$$R_r(\tau) = \sigma_n^2 g_{\sqrt{2}\sigma}(\tau) \ , \tag{14}$$

while the autocorrelation function of the noise convolved with the first derivative of a Gaussian kernel is

$$R_{r'}(\tau) = -R''_r(\tau) = -\sigma_n^2 g''_{\sqrt{2}\sigma}(\tau) \ . \tag{15}$$

Thus, the variances of the respective stochastic processes are

$$\begin{aligned} \sigma_r^2 &= R_r(0) = \frac{\sigma_n^2}{2\sqrt{\pi}\sigma} & (16) \\ \sigma_{r'}^2 &= R_{r'}(0) = \frac{\sigma_n^2}{4\sqrt{\pi}\sigma^3} \ . & (17) \end{aligned}$$

With this, the problem of determining the variance of the line position can be transformed to the equivalent problem of determining the variance of the zero crossing of an arbitrary function with exactly one zero crossing, e.g., (3), corrupted by a the addition of a stationary stochastic process with zero mean and a given autocorrelation function, e.g., (17). Without loss of generality, the zero crossing of the function can be assumed to be at the origin.

Before an approximation of the variance of a zero crossing with added noise will be derived, it is useful to look at this problem in a purely geometrical manner to get an intuitive notion of the dependency. Assume for the moment that the uncorrupted signal is simply the linear function $f(x) = hx$, and that the distribution of the noise $n(x)$ is uniform in some interval $[-m, m]$ for all x. Then, conceptually one can think of the corrupt signal $i(x)$ to be contained entirely within a "tube" of diameter $dy = 2m$ around $f(x)$, as shown in Figure 3. Therefore, the zero crossings must all lie in an interval of diameter dx around 0. Obviously, $dy/dx = h$, and therefore the width of this interval is proportional to $1/h = 1/f'(0)$. Although the distribution of the zero crossings within the interval is unknown, the variance of the zero crossings can be expected to be proportional to $1/f'(0)^2$. Thus, it can be conjectured that for general functions the variance of the zero crossings is proportional to $\sigma_n^2/f'(0)^2$.

In order to prove this conjecture, a Taylor series expansion of order one of (13) can be performed, where for the moment it should be assumed that $r_f(x)$ and $r_n(x)$ are not obtained by convolution with a Gaussian, but are an arbitrary function and noise with a given autocorrelation function, respectively. Thus, $r(x_0) = 0$ for some x_0, and hence

$$r(x_0) \approx r(0) + r'(0)x_0 = r_f(0) + r_n(0) + (r'_f(0) + r'_n(0))x_0 \ . \tag{18}$$

By assumption, $r_f(0) = 0$, and hence

$$x_0 = -\frac{r_n(0)}{r'_f(0) + r'_n(0)} \ . \tag{19}$$

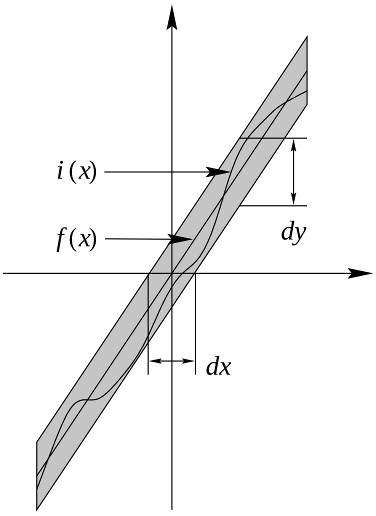


Figure 3. If the noise $n(x)$ is assumed to be uniformly distributed in $[-m, m]$ for all x and $f(x) = hx$, the zero crossings of $i(x) = f(x) + n(x)$ must be contained in an interval of width $dx = dy/h$.

The calculation of the variance of x_0 from (19) is extremely difficult. However, if one assumes that the variance $\sigma^2_{r'}$ of $r'_n(0)$ is small with respect to the first derivative of the signal $r'_f(0)$, the term $r'_n(0)$ can be omitted from the denominator of (19):

$$x_0 \approx -\frac{r_n(0)}{r'_f(0)} \quad . \tag{20}$$

With this simplification it is a simple matter to calculate the variance of x_0. The denominator is a known deterministic function, while the nominator is a stationary stochastic process with known autocorrelation $R_r(\tau)$, and therefore with known variance $\sigma^2_r = R_r(0)$. Thus, the variance σ^2_0 of the location x_0 of the zero crossing is given by

$$\sigma^2_0 = \frac{\sigma^2_r}{r'_f(0)^2} \quad . \tag{21}$$

It is useful to check whether this result also holds in practice. To do this, various functions $r_f(x)$ with exactly one zero crossing were sampled at discrete points x_i, $i = -32, \ldots, 32$. For the tests the following functions were used with varying signal levels h: $f(x) = hx$, $f(x) = h \arctan x$, $f(x) = h\sqrt{2\pi}(\phi(x) - 0.5)$, and $h \tanh(x)$. Note that all these functions have $r'_f(0) = h$. The noise $r_n(x)$ was generated by convolving Gaussian white noise of standard deviation 1 with a Gaussian smoothing kernel of width σ. Thus, it has the autocorrelation $R_r(\tau) = g_{\sqrt{2}\sigma}(\tau)$, i.e., $\sigma_r = 1/(2\sqrt{\pi}\sigma)$. For the experiments, $\sigma = 2$ was used. From the sampled values $r_i = r_f(x_i) + r_n(x_i)$, zero crossings are extracted to subpixel accuracy by linear interpolation. Since the relation $r'_f(0) = h$ does not hold for the sampled r_i, care is taken to choose h in such a way that the discrete approximation of the first derivative has the value h, i.e., $(r_1 - r_{-1})/2 = h$. For each function, 100000 experiments were performed, and the variance of the closest zero crossing to 0 was calculated from these experiments. The results are shown in Figure 4. As can be seen, for high signal to noise ratios, i.e., for large h, the variance of the zero crossings predicted by (21) and the experimentally determined variance match almost perfectly. If the signal level h is less than

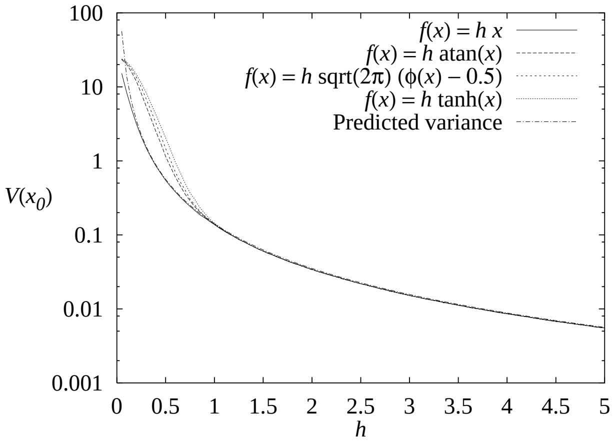


Figure 4. Variance of the zero crossing of different test functions when corrupted with noise of known autocorrelation and analytically predicted zero crossing variance. Note the logarithmic scale of the *y*-axis.

1, i.e., if the standard deviation of the noise is larger than approximately 40% of the amplitude of the signal, the term $r'_n(0)$ neglected in (20) becomes important. As can be seen, in this range of signal to noise ratios, the variance of the zero crossings is estimated too small by (21). Fortunately, such low signal to noise ratios occur extremely rarely in real images. One notable exception are Synthetic Aperture Radar (SAR) images, where the speckle effect can create very low signal to noise ratios [11]. Thus, (21) in most cases gives a very accurate estimate of the true variance of the zero crossings of a function corrupted by noise. If a more precise estimate of the variance is needed, one should note that this problem is intimately related to the problem of determining the first-passage density of a stochastic process to an arbitrary boundary, for which a solution was presented in [5]. However, since the first-passage density derived in [5] is rather complex and can only be calculated numerically, it is not applied here.

3.3: Variance of line and edge positions

With these preliminaries, the predicted variance of the line and edge positions can be determined easily. Since the edge model is much simpler than the line model it is considered first. To derive the variance of the edge locations in the 2D case, without loss of generality it can be assumed that the edge is straight and lies on the y-axis. This can be done because the independence of the unbiasedness of the edge and line positions on the orientation of the edge or line was shown for synthetic noise-free images in [17] and because the noise is assumed to be wide-sense stationary white noise, i.e., it is rotationally invariant. Therefore, the edge position is given by the zero crossings of the model edge convolved with the second directional derivative along the x-axis of a Gaussian kernel. Thus, the edge location is given by

$$r_{xx}(x, y) = r_{f,xx}(x, y) + r_{n,xx}(x, y) = 0 \ . \tag{22}$$

The 2D autocorrelation function $R_{r_{n,xx}}$ of $r_{n,xx}(x, y)$ is given by [18]

$$R_{r_{n,xx}}(\tau_x, \tau_y) = \sigma_n^2 g''''_{\sqrt{2}\sigma}(\tau_x) g_{\sqrt{2}\sigma}(\tau_y) \ , \tag{23}$$

where σ_n^2 is the variance of the noise in the original image. Thus, the variance of the random noise field is

$$\sigma_{r_{n,xx}}^2 = \frac{3\sigma_n^2}{16\pi\sigma^6} \ , \tag{24}$$

where σ is the standard deviation of the Gaussian smoothing kernel. For the response of the filter to the model step edge needed in (21), obviously

$$r_{f,xxx}(x, y) = hg''_{\sigma}(x) \ , \tag{25}$$

where h is the contrast of the edge. Thus,

$$r_{f,xxx}(0, 0)^2 = \frac{h^2}{2\pi\sigma^6} \ . \tag{26}$$

Therefore, the variance of the location of an edge in a 2D image is given by

$$\sigma_e^2 = \frac{\sigma_{r_{n,xx}}^2}{r_{f,xxx}(0, 0)^2} = \frac{\dfrac{3\sigma_n^2}{16\pi\sigma^6}}{\dfrac{h^2}{2\pi\sigma^6}} = \frac{3}{8}\frac{\sigma_n^2}{h^2} \ . \tag{27}$$

This result is quite surprising since it signifies that the variance σ_e^2 of the edge locations is independent of the choice of the smoothing parameter of the Gaussian kernel used to calculate the gradient of the image. This counterintuitive result should be verified by experiments. To do so, test images of size 32×32 were generated with exactly one edge in the center of the image. Note that the test images thus generated are mainly intended to verify (27) and to test whether it holds for discrete images. Real images will be considered in Section 4.3. From these images, edges were extracted with Gaussian derivatives of varying standard deviation σ. The input images were corrupted with Gaussian noise of varying standard deviation σ_n. For each combination of σ and σ_n, 1000 experiments were performed, resulting in roughly 30000 edge points from which the mean value and variance of the edge positions were computed. Figure 5 shows that the mean value of the edge positions is unbiased. The maximum deviation of the edge positions in this series of experiments is roughly one hundredth of a pixel. More importantly, Figure 6(a) shows the calculated variances of the edge positions. As can be seen, the results do not depend on the degree of smoothing. To check the quality of the edge position variance predicted by (27), Figure 6(b) displays the ratio of the predicted and extracted edge position variances. This ratio is a measure of the bias of (27). As can be seen, it is very close to 1 for most combinations of σ and σ_n. The maximum error occurs if σ and σ_n are close to zero. Overall, (27) gives an excellent estimate of the edge position variance.

In order to predict the variance of the line position in 2D images if the bias removal, i.e., the width and position correction, is not applied, the same technique can be used. As above, without loss of generality the line can be assumed to lie on the y-axis. Thus, the line position is given by the zero crossings of the model line convolved with the first directional derivative along the x-axis of a Gaussian kernel. Hence, the line location is given by

$$r_x(x, y) = r_{f,x}(x, y) + r_{n,x}(x, y) = 0 \ . \tag{28}$$

The 2D autocorrelation function $R_{r_{n,x}}$ of $r_{n,x}(x, y)$ is

$$R_{r_{n,x}}(\tau_x, \tau_y) = -\sigma_n^2 g''_{\sqrt{2}\sigma}(\tau_x) g_{\sqrt{2}\sigma}(\tau_y) \ , \tag{29}$$

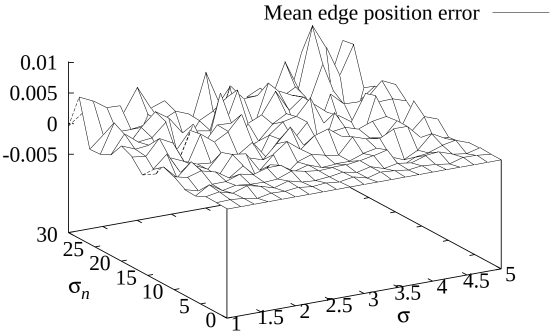
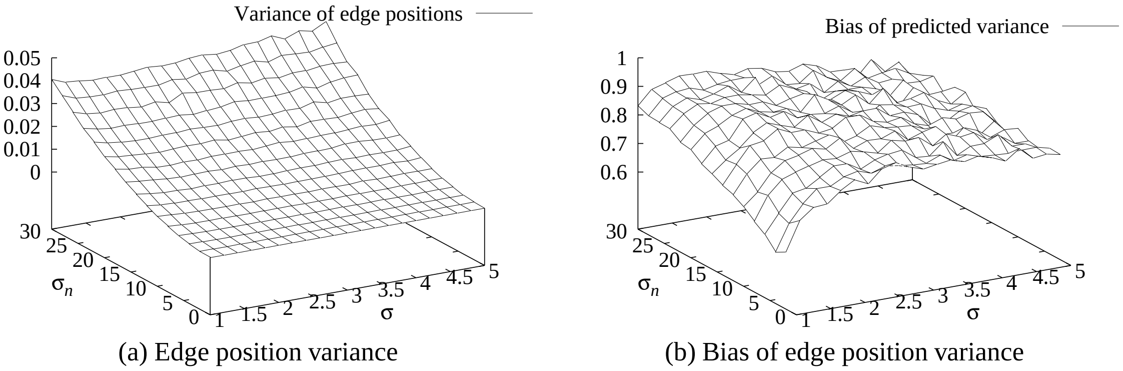


Figure 5. Mean value of the edge position as a function of the standard deviation σ of the Gaussian smoothing kernel and the standard deviation σ_n of the input noise.



(a) Edge position variance

(b) Bias of edge position variance

Figure 6. Variance of the edge position as a function of the standard deviation σ of the Gaussian smoothing kernel and the standard deviation σ_n of the input noise (a). Ratio of the predicted and extracted edge position variances, i.e., bias of the prediction.

and hence the variance of the random noise field is

$$\sigma^2_{r_{n,x}} = \frac{\sigma_n^2}{8\pi\sigma^4} \quad . \tag{30}$$

Since the asymmetry of a line causes the line position to shift, the denominator of (21) has to be evaluated at the biased line position l given by (5). With this, the variance of the position of a 2D line is given by

$$\sigma_l^2 = \frac{\sigma^2_{r_{n,x}}}{r_{f,xx}(0,0)^2} = \frac{\sigma_n^2}{8\pi\sigma^4}\frac{1}{h^2 r''_a(l,\sigma,w,a)^2} \quad , \tag{31}$$

where $r''_a(l,\sigma,w,a)$ is given by (4). Unfortunately, this expression cannot be simplified very much. Therefore, to get an impression of the dependency of the line position variance on different parameters, Figure 7(a) shows the predicted variance as a function of the standard deviation σ of the Gaussian smoothing kernel, while Figure 7(b) shows the predicted variance as a function of the line width w. As can be seen, the line position variance is smallest for $\sigma = w$. Furthermore, if σ is chosen much too small for a given line width w, the line position variance increases rapidly.

Again, it has to be checked whether (31) holds for synthetic test images. To do so, images of size 32×32 were generated with exactly one line lying in the center of the image. Note again that the

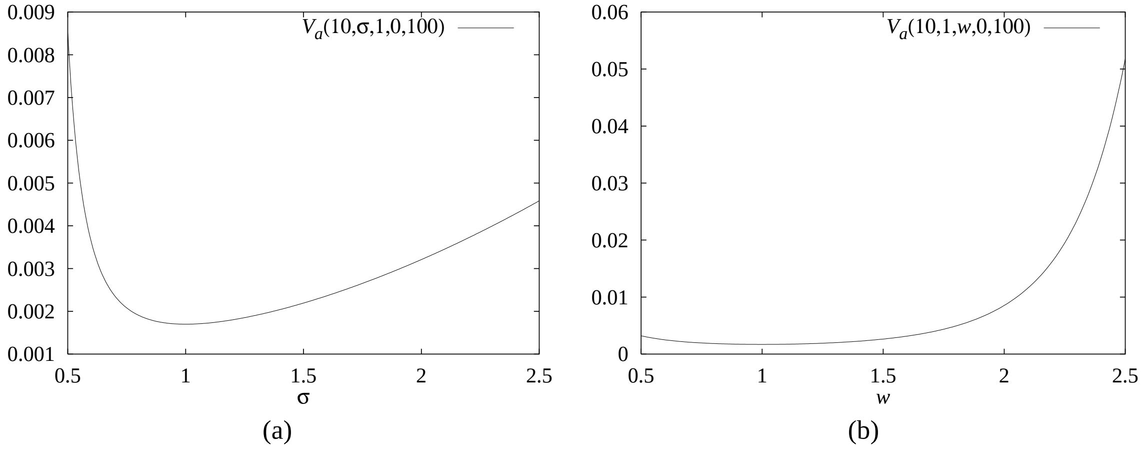


Figure 7. Predicted variance of the line position as a function of the standard deviation σ of the Gaussian smoothing kernel for a line of width *w* $=$ 1, contrast *h* $=$ 100, and noise of standard deviation $\sigma_n = $ 10 (a), and as a function of the line width *w* for $\sigma = $ 1, *h* $=$ 100, and $\sigma_n = $ 10 (b).

test images thus generated are mainly intended to verify (31) and to test whether it holds for discrete images. Real images will be considered in Section 4.2. To simulate the typical application case, the σ used for smoothing was kept constant, while the total width $2w$ of the line was varied in the interval $[\sigma, 5\sigma]$ in subpixel increments. To obtain subpixel line widths in the image, it was assumed that the sensor has a square aperture function, i.e., integrates the incoming light intensity over the area of each pixel. Therefore, the intensity of a pixel is proportional to the area of the pixel covered by the line. Furthermore, the asymmetry a of the line was varied in the interval $[0, 0.75]$. The input images were corrupted with Gaussian noise of varying standard deviation σ_n. For each combination of a, w, and σ_n, 1000 experiments were performed, resulting in roughly 30000 edge points from which the mean value and variance of the edge positions were computed. For the experiments, lines of contrast $h = 100$ were extracted with $\sigma = 2$. Figure 8 displays the mean line position error, i.e., the difference between the mean value of the line positions extracted from the image and the true line positions for $a = 0$. As can be seen, for all integer line widths the mean error is very close to zero. For non-integer line widths the mean may lie up to 0.07 pixels from the true line position. This effect is caused by extrapolation errors in the subpixel extraction, and not by noise, as is discussed in [17]. Thus, noise introduces no new bias of the line positions, and therefore the line detector can be regarded as unbiased in the presence of noise. Figure 9(a) shows the extracted line position variance. As can be seen, the cross sections of the surface in this figure closely resemble the shape of the predicted variance displayed in Figure 7(b). To check the validity of (31), Figure 9(b) shows the ratio of the predicted and extracted line position variances. This measure of the bias of (31) is very close to 1 for almost all combinations of w and σ_n. For integer line widths and small noise variances σ_n^2, the extracted line position variance is larger than the predicted variance by up to a factor of five because the line positions lie at the borders of a pixel, where the extrapolation error is largest, and thus one should expect the variance to be larger in these cases. Similar results are also obtained for all other values of the asymmetry a. Figure 10 exemplifies this by showing the line position variance and its bias for $a = 0.5$. As can be seen, the measure of the bias is again very close to 1 for almost all combinations of w and σ_n. Overall, (31) gives an excellent estimation of the line position variance for uncorrected line extraction results.

Another very important performance measure for lines is the variance of the extracted line

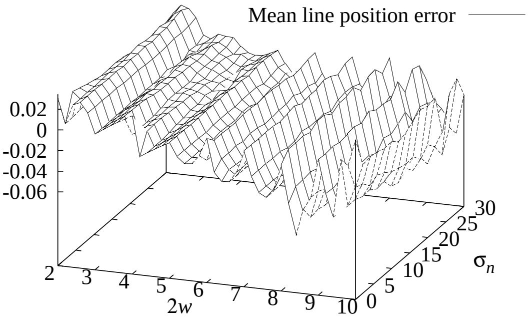
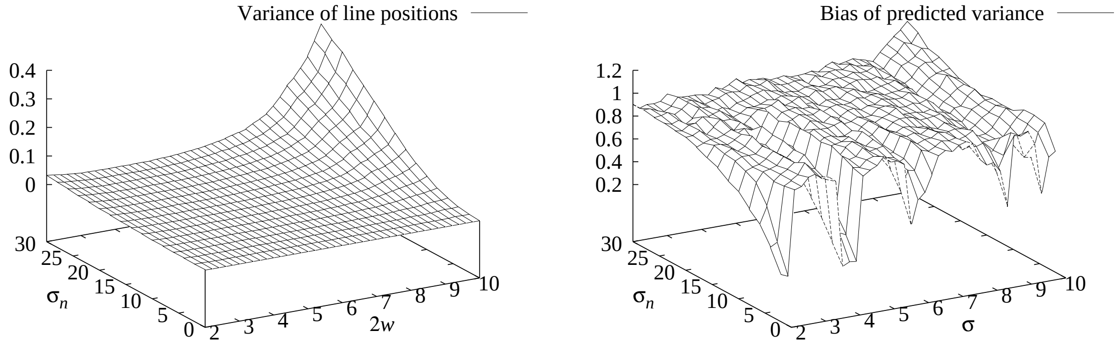


Figure 8. Mean value of the line position as a function of the total line width 2*w* and the standard deviation σ_n of the input noise.

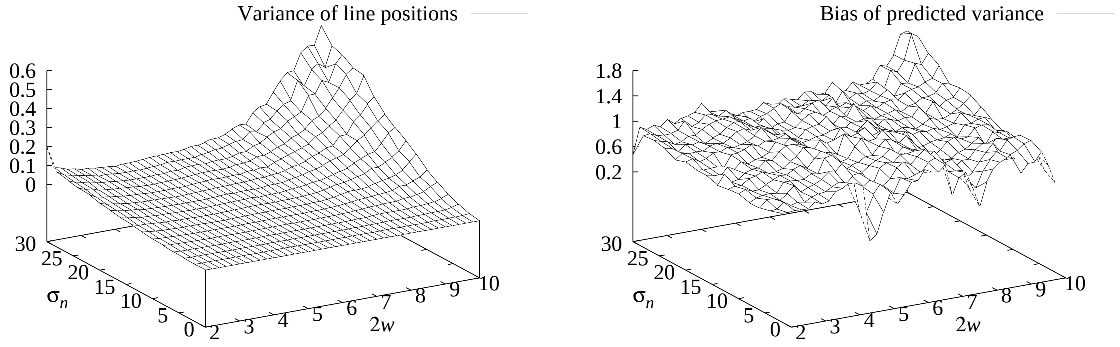


(a) Line position variance

(b) Bias of line position variance

Figure 9. Variance of the line position as a function of the total line width 2*w* and the standard deviation σ_n of the input noise (a). Ratio of the predicted and extracted line position variances, i.e., bias of the prediction (b).

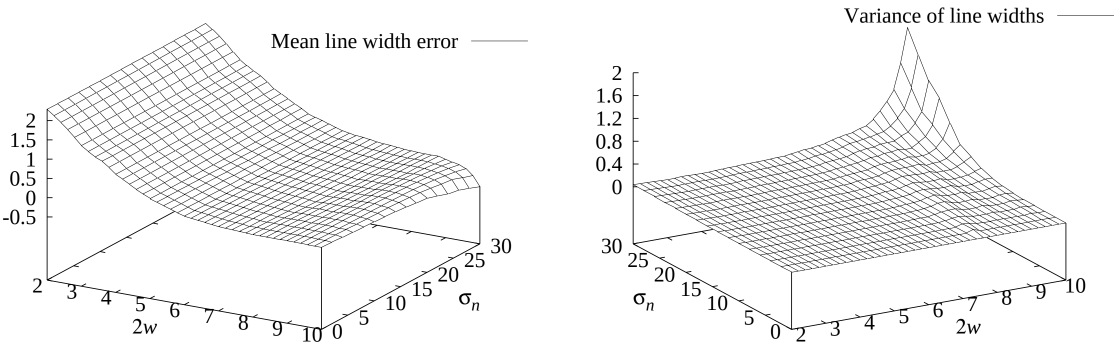
widths. Since the total line width in case the width correction is not applied is given by the distance of the two edges of the line, in principle it seems possible to describe the variance of the line width using the approach introduced in this section. For example, one could calculate the variances of the two edge positions, assume they are independent, and obtain the variance of the line width as the sum of the two edge position variances. However, this has several problems. First, the edge locations at which (21) must be evaluated can only be calculated numerically. Therefore, no analytical result is possible. Instead, the width variance would have to be tabulated. More importantly, however, the edge positions, and thus their variances, are not independent of one another, as can be seen from the discussion in Section 2. Finally, since the first edge point found on each side of the line is used to calculate the width of the line without taking the edge strength into account, it can be expected that for high noise levels the edge position is much more affected by noise than predicted by an analytical model because then the first edge point found may be caused by noise instead of the true edge point. For these reasons, the approach of adding the variances of the two edge locations cannot return very good results, and this was verified on the synthetic examples used above for the line position variance. From this discussion it follows that a good analytical prediction of the variance of the line width seems very hard to obtain. Therefore, this topic has not been pursued further in this paper. Instead, only the experimentally obtained means and variances of the line width will be discussed. These are shown for $a = 0$ in Figures 11(a) and (b), respectively.



(a) Line position variance

(b) Bias of line position variance

Figure 10. Variance of the line position for $a = 0.5$ (a). Bias of the variance prediction for $a = 0.5$ (b).



(a) Mean line width error

(b) Line width variance

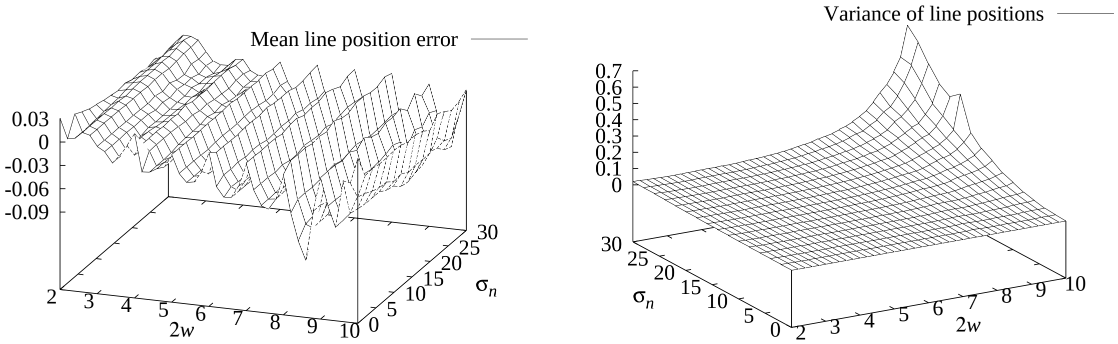
Figure 11. Mean (a) and variance (b) of the line width as a function of the total line width 2*w* and the standard deviation σ_n of the input noise.

As can be seen from Figure 11(a), the noise has almost no influence on the extracted mean width error, i.e., the difference of the mean line width and the true line width. As is to be expected from the discussion in Section 2, the mean line width is biased. Figure 11(b) displays the experimentally obtained variance σ_w^2 of the line widths. As can be seen, it again increases proportionally to σ_n^2. Furthermore, the width variance is relatively small for small w. It increases significantly only for large w. Again, similar results hold for all asymmetries a.

The final important question is how the bias removal, i.e., the position and width correction, influences the variances of the line position and width. Conceptually, the bias removal is given by a function $f : (v, r) \mapsto (w, a)$, where v is the width extracted from the image, r is the gradient ratio at the two extracted edge points, w is the true line width, and a is the true asymmetry. Thus, standard error propagation schemes could be used to propagate the variances σ_l^2 and σ_w^2 through f. If the covariance $\sigma_{l,w}$ of the line position and width were known, the covariance matrix Σ_u of the uncorrected could be calculated, and the covariance matrix Σ_c of the corrected line position and width would simply be given by [9]

$$\Sigma_c = Df \Sigma_u Df^T \ , \tag{32}$$

where Df is the Jacobian matrix, i.e., the first derivative, of f. Since σ_w^2 is only known empirically, since an explicit estimation for $\sigma_{l,w}$ is very hard to obtain, and since f can only given in tabulated



(a) Mean line position error

(b) Line position variance

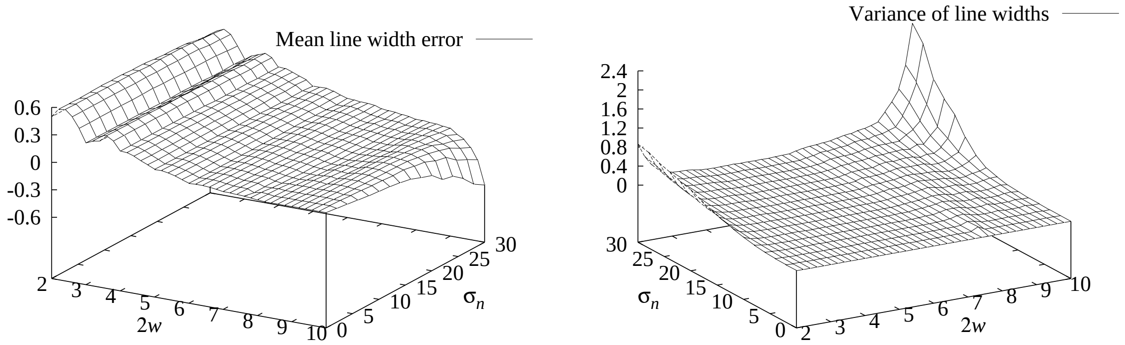
Figure 12. Mean (a) and variance (b) of the line position as a function of the total line width 2*w* and the standard deviation σ_n of the input noise if the line width and position correction is applied.

form, no attempt has been made to derive an explicit formula for Σ_c. However, from the shape of the bias inversion function displayed in [15, 16, 17] it can be estimated that the variances of the corrected line position and width will be slightly larger than the uncorrected variances because the partial derivatives of f are larger than 1 almost everywhere. Figure 12 displays the mean and variance of the corrected line position, while Figure 13 shows the mean and variance of the corrected line width. From Figure 12(a) it can be seen that the line position correction slightly increases the mean error in areas of non-integer line width, i.e., in areas where the line position does not lie close to the center of a pixel. The maximum error increases from approximately 0.07 to 0.09 pixels. Furthermore, from Figure 13(a) it can be seen that the corrected line widths are much closer to the true line widths. The corrected line widths exhibit a small bias only if the line width is very small compared to σ. This fact was also observed for noiseless images in [17], and thus comes as no surprise. The variances of the line position are slightly decreased for small w, while they are slightly increased for large w, as can be seen when Figure 12(b) is compared to Figure 9(a). On the other hand, the line width variance increases by a large amount for small w, while the increase is small for large w, as can be seen by comparing Figures 13(b) and 11(b). Essentially, for small w the correction seems to trade better line position variance for worse line width variance. As above, qualitatively similar results hold for all other values of the asymmetry a.

In summary, from the discussion in this section it can be concluded that the extracted line positions and widths are unbiased in the presence of noise, and that the variances of the line positions and widths are very small. Thus, subpixel accuracy can definitely be achieved for noisy synthetic images, while it seems very likely that this kind of accuracy may also be achievable in real images.

To conclude this section, it should be noted that sometimes other performance measures apart from the localization quality also play an important role. The most often used measure is the detectability of a feature in the presence of noise. Essentially, this can be modeled by the signal to noise ratio, i.e., the ratio of squared magnitude of the derivative of the model feature in which the thresholding is done and the variance of the noise in this derivative. For edge detection, the relevant derivative is the first derivative, while for line detection it is the second derivative. Thus, for edge detection the signal to noise ratio r is given by

$$r = \frac{r_e'(0,\sigma,h)^2}{\sigma_{r_{n,x}}^2} = \frac{4h^2\sigma^2}{\sigma_n^2} \ , \tag{33}$$



(a) Mean line width error

(b) Line width variance

Figure 13. Mean (a) and variance (b) of the line width as a function of the total line width 2*w* and the standard deviation σ_n of the input noise if the line width and position correction is applied.

while for lines it is given by

$$r = \frac{h^2 r_a''(l,\sigma,w,a)^2}{\sigma_{r_{n,xx}}^2} = \frac{16\pi\sigma^6 h^2 r_a''(l,\sigma,w,a)^2}{3\sigma_n^2} \quad . \tag{34}$$

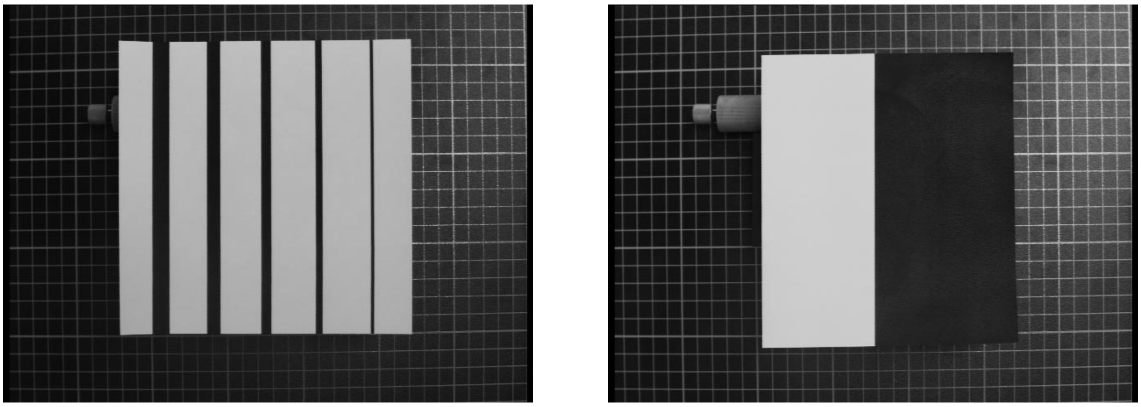
This leads to the fairly obvious conclusion that features become easier to detect as the standard deviation σ of the Gaussian used to smooth the image is increased. Of course, this only holds if there are no other nearby features complicating the extraction.

4: Experiments on real images

All the experiments so far have been done on synthetic test images. Since for real imaging conditions various assumptions made in the test so far may not hold, it is important to perform experiments on real images. For example, the camera used to acquire an image may not have an ideal square aperture function or the framegrabber may not exhibit a linear increase in gray values as the scene intensity increases linearly.

4.1: Experimental set-up

In order to test the subpixel accuracy of the line detection algorithms the following strategy was adopted: A test set containing five lines of widths 2 mm–10 mm were printed on a high-quality laser printer. The resulting print was mounted on a table which can be shifted in one direction in 10 μm increments. A camera with a 12 mm lens was mounted perpendicular to the table at a distance of approximately 50 cm. A standard analog framegrabber was used to acquire the images, since this is by far the most common setup in industrial applications, where subpixel measurements are extremely important. For edge detection, the test set contained a single edge in the center of the print. These test images were chosen because they are representative for many industrial inspection tasks, where often manufactured workpieces having straight or moderately curved edges or lines have to be measured. Of course, this test set-up is by no means intended to be exhaustive. Figure 14 shows an image of the line and edge test sets. Note that, although the lens used in this test is fairly good for computer vision standards, there is nevertheless a significant radial distortion. To test the subpixel accuracy, each of the test sets was shifted by 1 mm in increments of 50 μm, resulting in



(a) Line test set

(b) Edge test set

Figure 14. Test sets used to check the subpixel accuracy of the line and edge detection algorithms.

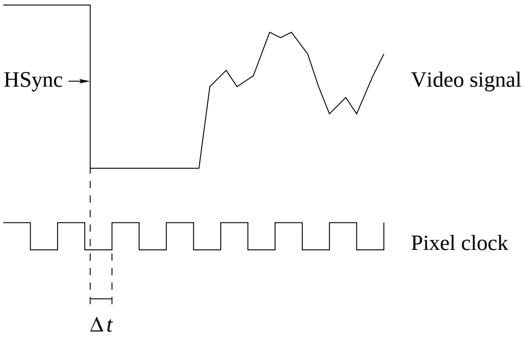
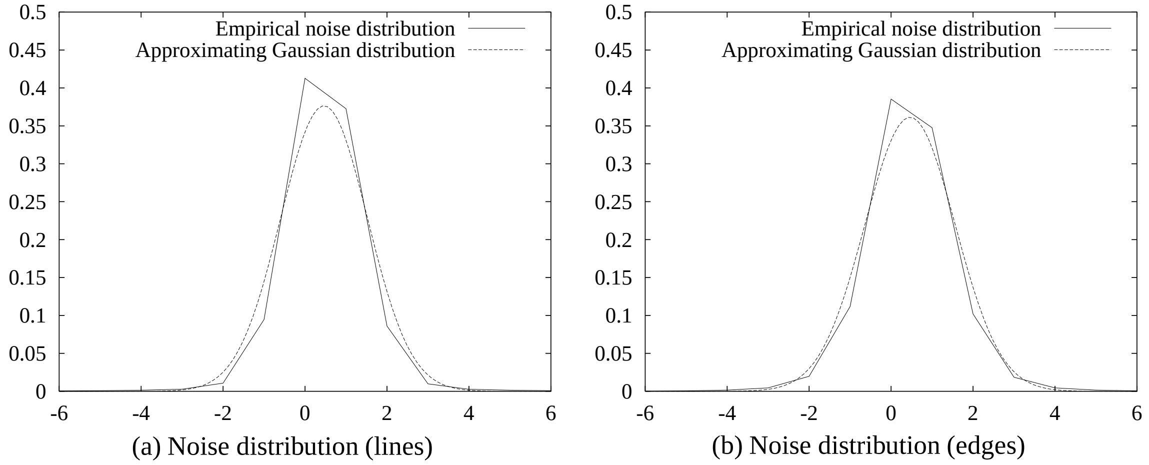


Figure 15. Scheme to start the digitization of one line of a video signal in current framegrabber architectures: Digitization is started at the first rising flank of the unsynchronized pixel clock after the horizontal sync signal has occurred, resulting in an offset of Δt.

a shift of approximately 2 pixels in total in the image. In each position of the measurement table, 20 images were taken. Thus, for each test set a total of 420 images were taken. The goal of the test is to determine the precision, i.e., the variance, of the line and edge position measurements and the absolute position shift of the line and edge positions in order to decide whether it is possible to detect subpixel shifts of 50 μm or, equivalently, approximately one tenth of a pixel.

Before this can be done, a few words on the architecture of standard analog framegrabbers are necessary since the technology used for them will influence the quality of the acquired images. The main implementation issue for framegrabbers is when to start digitizing a line of the video signal. To do this, most analog framegrabbers have a clock chip running at several times the pixel resolution, typically 4–10 times. They start the digitization process on the first upward flank of the pixel clock signal after the falling flank of the horizontal synchronization (HSync) signal of the video signal has occurred, as shown in Figure 15. Thus, for each line there is a random offset of Δt by which the line is shifted with respect to the true HSync signal, resulting in a random shift by up to one fourth to one tenth of a pixel, depending on the frequency of the pixel clock. This seems to prevent subpixel accuracy of better than one tenth of a pixel. Fortunately, however, the



(a) Noise distribution (lines)

(b) Noise distribution (edges)

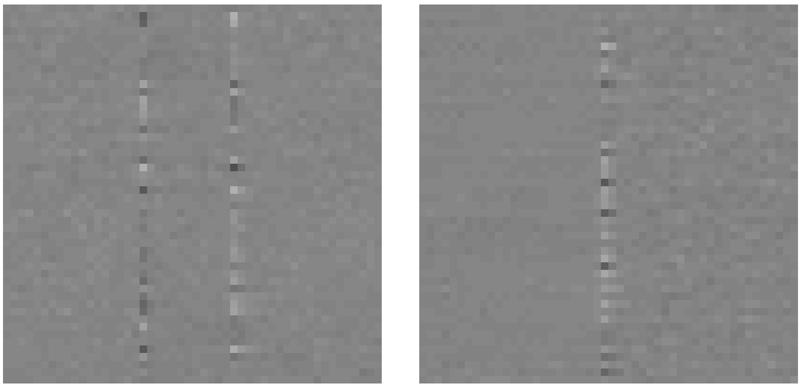
Figure 16. Experimentally determined noise distributions and Gaussian distributions with the same mean and variance for the line test set (a) and the edge test set (b).

random offset is independent for each line of the video signal. Thus, one can expect to achieve better accuracy by the smoothing done in the line or edge detector. In effect, the random shifts in the digitized video signal can be modeled as an additional noise source. From this discussion it follows that the noise should be larger at the border of the features of interest since the subpixel shifts result in larger gray value variations there, while they have very little influence in regions of approximately constant gray value. Of course, this problem does not occur if the framegrabber is synchronized with the pixel clock of the camera or if a digital camera is used.

In order to test the hypothesis that the random shifts in each line of the video signal lead to increased noise at edge positions and to obtain the noise statistics of the test images, the average of all 20 images within each shift can be calculated. Then, the individual images can be subtracted from the mean image to obtain the noise distribution for that particular image. If the distributions over all 420 images are averaged, a very good approximation of the noise distribution is obtained. Figure 16 shows the noise distributions thus obtained for the line and edge test set, superimposed by a Gaussian distribution of the same mean and variance. As can be seen, the Gaussian distribution yields a very good approximation of the noise distribution in both cases. The standard deviations of the noise were estimated as approximately 1.06 and 1.10, respectively, which is very small. Figure 17 shows small parts of test images subtracted from their corresponding mean image. As can be seen, the random shifts lead to significantly increased noise at the edges of the line in Figure 17(a) and at the edge in Figure 17(b). Therefore, the line and edge locations may not be as accurate as one would expect from the discussion in the previous section.

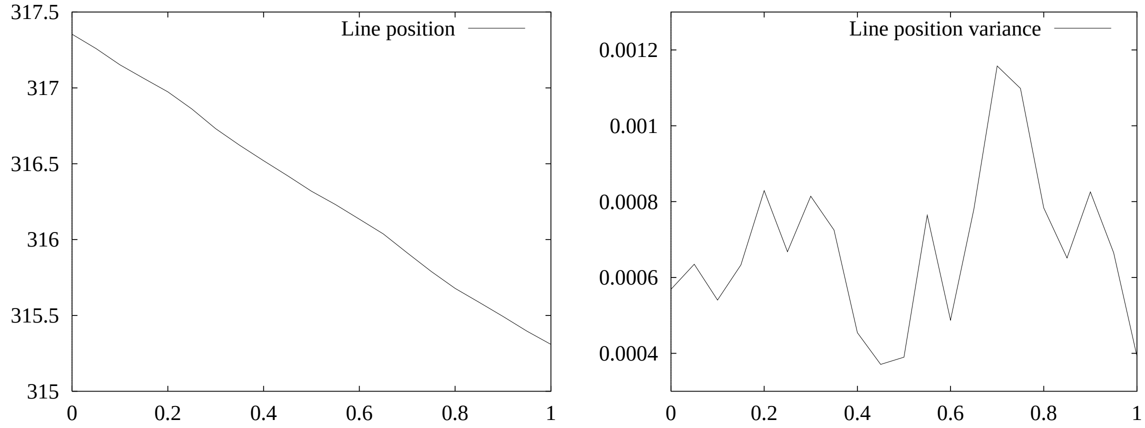
4.2: Subpixel accuracy of line position and width

To test the subpixel accuracy of the line detection algorithm, only the center line of the test set of width 6 mm was used since it turned out that the lens distortion was too large to obtain meaningful results for the other four lines. Although the center line was carefully aligned to be as vertical as possible, even for this line the lens distortion influences the extraction results, if statistics are calculated over the entire line. Therefore, small windows of height 25 were used to calculate the mean line position and width, and their corresponding variances, because for these small windows



(a) Noise in a line image (b) Noise in an edge image

Figure 17. Examples of a test image subtracted from its corresponding mean image for the line test set (a) and the edge test set (b). The gray values in the images have been scaled by a factor of 5 for clarity.



(a) Line position (b) Line position variance

Figure 18. Line position and variance as a function of the shift of the line in mm.

the lens distortion played only a small role. Since the means and variances are calculated over 20 images, a total of 500 points are used to obtain the statistics. The experiments showed that the resulting statistics are independent of the position of the window. Because the center line of the test images has a total width of approximately 12 pixels, the optimal $\sigma = 6/\sqrt{3}$ [16] was used for the extraction. Figure 18(a) shows the extracted line position as a function of the shift of the line in mm. The resulting curve is almost a perfect straight line. The precision of the line position, i.e., its variance, is displayed in Figure 18(b), from which it can be seen that the maximum variance is approximately 0.00115, i.e., the maximum standard deviation is roughly one thirtieth of a pixel. From (31) the line position variance can be estimated as $1.84 \cdot 10^{-5}$. Thus, the extracted position variance is larger than the predicted variance by a factor of 62.5 mainly due to the random line shifts induced by the framegrabber, the still noticeable lens distortion in this small part of the image, and the fact that the line is not perfectly aligned with the vertical axis. However, since the position variance is still very small it can be expected that the goal of these experiments to detect subpixel shifts of one tenth of a pixel can be achieved. This can easily be verified by testing the hypothesis

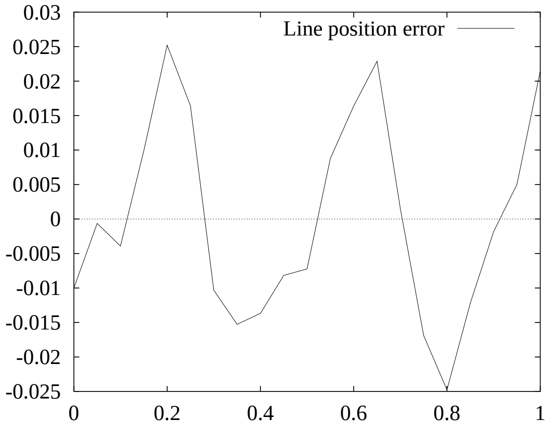
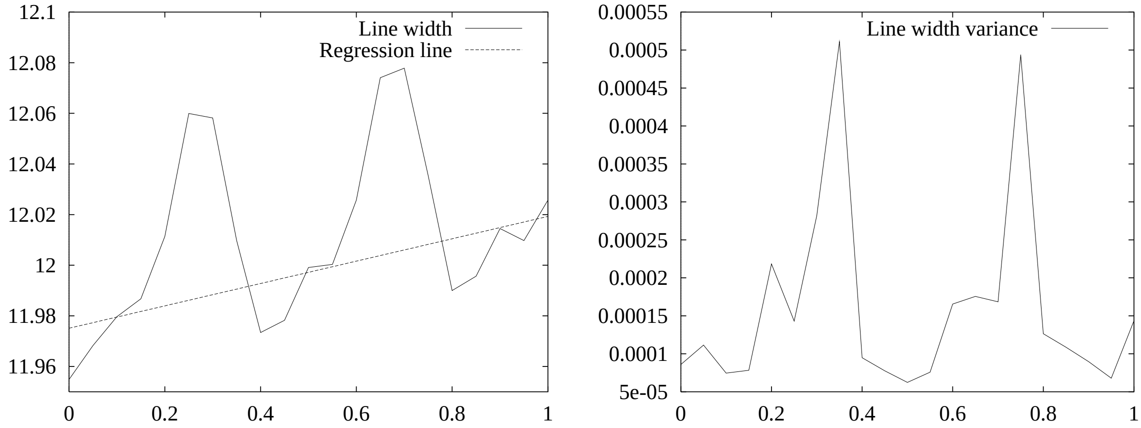


Figure 19. Absolute error of the line position calculated as the difference of the extracted line positions and their regression line as a function of the shift of the line in mm.

that the line positions corresponding to two adjacent shifts are equal [6]. These tests have been performed and the hypothesis that the line positions are equal can be rejected with a probability greater than 99.9% for all line positions. Therefore, relative shifts of one tenth of a pixel can definitively be detected in real images.

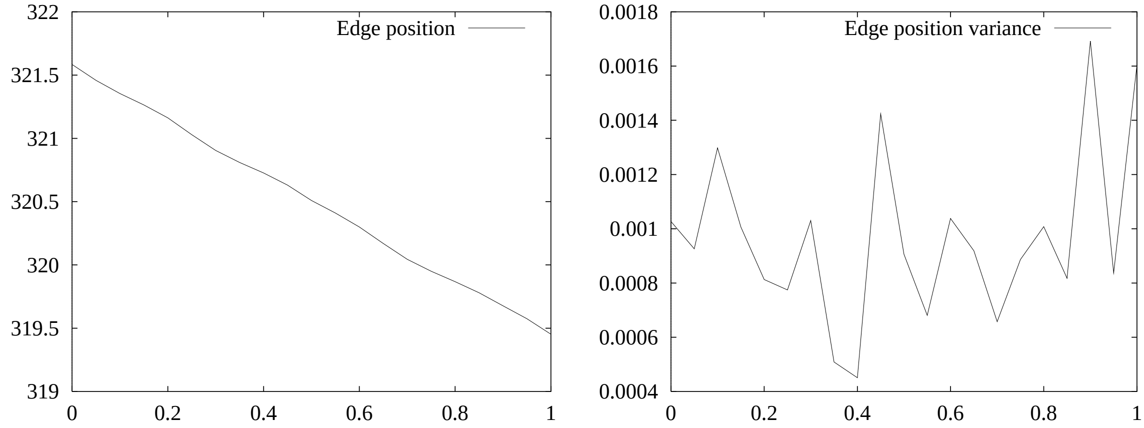
Another interesting point is the absolute position accuracy of the line. Since the camera was not calibrated, this is very hard to obtain. If, however, it is assumed that the linear shift of the object in the real world corresponds to a linear shift of the line in the image, a straight line can be fitted through the line positions extracted from the image. The fitting should, of course, take the extracted position variances into account. The equation of the line obtained for the experiment shown in Figure 18 is $-2.07633x + 317.364$. Therefore, it is likely that 1 mm in the real world corresponds to 2.07633 pixels in this part of the image. With this, the absolute position error of the line can be calculated as the difference of the extracted line position and the regression line. The results are shown in Figure 19. As can be seen, the absolute position errors are less than one fortieth of a pixel. The remaining errors show a systematic sinusoidal component, which may be caused by the mapping of the scene intensity to the gray values in the image performed by the framegrabber. Since they are very small, the line extraction algorithm yields very good absolute accuracy on real images as well. Of course, to get meaningful positions in real world coordinates the camera must be calibrated.

The final question for the line detection algorithm is the accuracy of the extracted line widths. The extracted line width and its variance are shown in Figure 20. The line width seems to be centered around 12 pixels. If the above discussion on the correspondence between 1 mm to 2.07633 pixels is taken into account, this would imply that the width of the 6 mm wide line is underestimated by approximately 3.7%. Therefore, the extracted line width is very close to the true line width. Furthermore, it can be seen by the regression line superimposed in Figure 20(a), that the line width increases as the shift increases. This can be attributed to the fact that for increasing shifts the line moves towards the center of the image where lens distortions are not as severe. Finally, Figure 20(b) displays the line width variance obtained from this experiment. As can be seen, the variance is less than 0.0005 almost everywhere, i.e., the standard deviation of the extracted line widths is less than one fortieth of a pixel. Thus, line widths can be extracted with very high subpixel accuracy.



(a) Line width (b) Line width variance

Figure 20. Line width and variance as a function of the shift of the line in mm.



(a) Edge position (b) Edge position variance

Figure 21. Edge position and variance as a function of the shift of the edge in mm.

4.3: Subpixel accuracy of edge position

To conclude this section, the results of performing the same kind of experiment for the subpixel edge detection algorithm are discussed. The edges were extracted using Gaussian derivatives with $\sigma = 1$. The resulting edge positions are exhibited in Figure 21(a). As was the case for lines, the extracted edge positions lie very close to a perfect straight line. Figure 21(b) displays the corresponding variances of the edge positions. The maximum variance is approximately 0.0017. The variance predicted by (27) is $1.57 \cdot 10^{-5}$, i.e., the extracted variance is larger than the predicted variance by a factor of 108. Again, this is caused by the random shifts of the discretized lines of the video signal, the lens distortion, and the fact that the edge is not perfectly aligned with the vertical axis. However, the standard deviation of the edge positions is still very small, being approximately one twenty-fifth of a pixel. Therefore, with the same hypothesis test as used above, it can be shown that edge shifts of one tenth of a pixel can be detected with better than 99.9% probability. Finally, as above, the absolute position error can be estimated by fitting a straight line through the extracted line positions. In this case, the equation of the line is $-2.12664x + 321.568$.

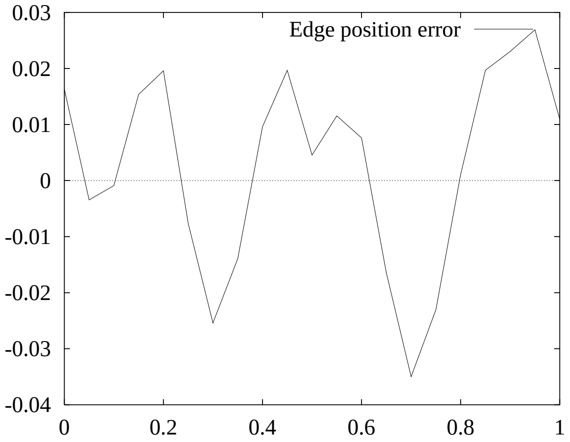


Figure 22. Absolute error of the edge position calculated as the difference of the extracted edge positions and their regression line.

The resulting absolute position errors are depicted in Figure 22. The maximum absolute error is approximately one thirtieth of a pixel. Therefore, edges can be extracted with very good absolute subpixel accuracy. If the remaining errors of up to one thirtieth of a pixel are too large for an application, a scheme as the one given in [14] could be used to estimate and remove the remaining errors. However, since the remaining errors are not generic, i.e., depend on the camera, lens, and framegrabber used, they are not modeled in this paper.

5: Conclusions

A thorough performance analysis is carried out for the line and edge extraction algorithms proposed in [15, 16, 17]. The quality of the line extraction results is evaluated for noisy images. Analytical formulas predicting the variance of the line and edge positions in case the variance of the image noise is known are derived and shown to correspond very well to experimentally determined line and edge position variances. Additionally, the variance of the line width in the presence of noise is determined experimentally. Finally, the quality of the line and edge positions and of the line widths is analyzed for real images. The results of this performance analysis indicate that the extraction results are unbiased for noisy synthetic and real images, that the variances of the extracted features are very small, i.e., the sub-pixel precision is very high, and that the absolute positions and widths can be determined to a very high degree of sub-pixel accuracy. For real images it is shown that position shifts of one tenth of a pixel can be detected with a probability of more than 99.9%, indicating that much better sub-pixel accuracy than one tenth of a pixel can be achieved for real images. Thus, it is shown that the line and edge extraction algorithms not only achieve sub-pixel resolution, but also sub-pixel precision and accuracy.

References

[1] Kalle Åström and Anders Heyden. Stochastic modelling and analysis of sub-pixel edge detection. In *13th International Conference on Pattern Recognition*, volume II, pages 86–90, 1996.

[2] Regine Brügelmann and Wolfgang Förstner. Noise estimation for color edge extraction. In W. Förstner and S. Ruwiedel, editors, *Robust Computer Vision: Quality of Vision Algorithms*, pages 90–106, Karlsruhe, 1992. Wichmann.

[3] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, June 1986.

[4] Didier Demigny and Tawfik Kamlé. A discrete expression of Canny's criteria for step edge detector performances evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1199–1211, November 1997.

[5] J. Durbin. The first-passage density of a continuous Gaussian process to a general boundary. *J. Appl. Prob.*, 22:99–122, 1985.

[6] Marek Fisz. *Wahrscheinlichkeitstheorie und mathematische Statistik*. VEB Deutscher Verlag der Wissenschaften, Berlin, 11th edition, 1989.

[7] Wolfgang Förstner. *Statistische Verfahren für die automatische Bildanalyse und ihre Bewertung bei der Objekterkennung und -vermessung*. Number 370 in Reihe C. Deutsche Geodätische Kommission, München, 1991.

[8] Wolfgang Förstner. 10 pros and cons against performance characterization of vision algorithms. In Henrik I. Christensen, Wolfgang Förstner, and Claus B. Madsen, editors, *Workshop on Performance Characteristics of Vision Algorithms*, pages 13–29, April 1996.

[9] Robert M. Haralick. Propagating covariance in computer vision. In Henrik I. Christensen, Wolfgang Förstner, and Claus B. Madsen, editors, *Workshop on Performance Characteristics of Vision Algorithms*, pages 1–12, April 1996.

[10] C. Heipke, H. Mayer, C. Wiedemann, and O. Jamet. Evaluation of automatic road extraction. In *International Archives of Photogrammetry and Remote Sensing*, volume XXXII, part 3–4W2, pages 151–160, 1997.

[11] Olaf Hellwich, Helmut Mayer, and Gerhard Winkler. Detection of lines in synthetic aperture radar (SAR) scenes. In *International Archives of Photogrammetry and Remote Sensing*, volume XXXI, part B3, pages 312–320, 1996.

[12] Jack Koplowitz and Vito Greco. On the edge location error for local maximum and zero-crossing edge detectors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(12):1207–1212, December 1994.

[13] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, Inc., New York, NY, 3rd edition, 1991.

[14] Frederico Pedersini, Augusto Sarti, and Stefano Tubaro. Estimation and compensation of subpixel edge localization error. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1278–1284, November 1997.

[15] Carsten Steger. Removing the bias from line detection. In *Computer Vision and Pattern Recognition*, pages 116–122, 1997.

[16] Carsten Steger. An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2), February 1998.

[17] Carsten Steger. *Unbiased Extraction of Curvilinear Structures from 2D and 3D Images*. PhD thesis, Fakultät für Informatik, Technische Universität München, 1998. Submitted.

[18] A. A. Sveshnikov. *Applied Methods of the Theory of Random Functions*. Pergamon Press, Oxford, 1966.

[19] Hemant D. Tagare and Rui J. P. deFigueiredo. On the localization performance measure and optimal edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1186–1190, December 1990.