SUBPIXEL-PRECISE EXTRACTION OF LINES AND EDGES

Carsten Steger
MVTec Software GmbH
Neherstraße 1, 81675 München, Germany
steger@mvtec.com

Working Group III/3

KEY WORDS: Edge Detection, Line Detection, Subpixel Precision, Subpixel Accuracy, Bias

ABSTRACT

Novel approaches to extract curvilinear structures, i.e., lines and edges, from 2D images are proposed. For lines, explicit geometric models for line profiles are used to analyze their behavior in scale-space. From this analysis, algorithms to extract lines and their widths with subpixel resolution are derived, and it is shown that the line positions and widths are necessarily biased by the smoothing used in the extraction. Since the mapping that describes the bias is invertible, it can be removed, leading to unbiased and hence accurate results. To extract edges, they are regarded as bright lines in the gradient image. Furthermore, by a scale-space analysis it is shown why line and edge junctions often cannot be extracted. From this analysis, a new method to extract complete junction information is derived. Finally, the line and edge extraction algorithms are extended to multispectral images.

1 INTRODUCTION

Lines and edges are important objects for many applications. They arise quite naturally as the low-level primitives of choice to extract from images. For example, roads, railroads, and rivers can be regarded as lines in aerial or satellite images. The extraction of these kinds of objects is very important for the data capture or update of geographic information systems (GIS) (Baumgartner et al., 1999, Wiedemann et al., 1998). Obviously, for these applications the width of the line is essential since it characterizes the type of a road, for example. Edges are also important in numerous other applications, e.g., building extraction (Bignone et al., 1996, Fischer et al., 1997) or object recognition (Lanser et al., 1997).

The approaches to line extraction can be classified into different categories (see (Steger, 1998b, Steger, 1998c) for a detailed review). In the first category, lines are regarded as ridges and ravines in the image (Eberly et al., 1994). Various differential geometric definitions can then be used to extract the ridges: as points on the contour line where the curvature is maximum (Maintz et al., 1996), as points where the principal curvature is a local maximum in the direction of maximum curvature (Armande and Montesinos, 1999), or as points where the gray values are restricted maxima or minima in the direction of the maximum eigenvalue of the Hessian (Busch, 1996). The first two definitions have the problem that ridges will often be extracted in the wrong position, even under very good conditions (Koenderink and van Doorn, 1994). Most notably, the first definition is unable to detect flat ridges, while the second definition will return a double response to a bar-shaped line if the image is not smoothed by a sufficient amount (Steger, 1998b). The third definition returns wrong line locations if the line has different lateral contrast, as we will see below. This definition can also be iterated through scale-space to obtain a coarse estimate of the line width (Lindeberg, 1998). The problem with this iterative scale-space approach is that at the optimal scale for width estimation the line position will in general be severely biased. None of the above approaches is able to extract lines that have a staircase profile.

In the second category, lines are extracted by using special filters. In (Koller et al., 1995) lines are regarded as objects that have two parallel edges and filters tuned in direction and scale are constructed to extract them. A similar approach is used in (Heitger, 1995), where directionally tuned filters are used for feature detection. Since these filters also respond to non-features, another set of directionally tuned filters is used to suppress the unwanted responses. Another very similar approach is used in (Iverson and Zucker, 1995), where the suppression is done by a non-linear combination of the filters. The problem with all these approaches is that since special filters need to be constructed for several or even all possible line directions, the extractors are computationally very expensive. Furthermore, the last two approaches do not extract the line width. The first approach obtains a coarse estimate of the line width by an iteration through scale-space, which makes the approach even more expensive. The last two approaches can also be used to extract edges.

A very interesting approach to the combined extraction of points, linear features (i.e., lines and edges), and blobs was proposed in (Förstner, 1994, Fuchs, 1998). The average squared gradient, i.e., the matrix $\Gamma f = g_{\sigma_2} * (\nabla(g_{\sigma_1} * f)\nabla(g_{\sigma_1} *$

$f)^T$), where f denotes the image, is used to classify the image into homogeneous and non-homogeneous regions. For multispectral images, the average squared gradients in the individual channels are weighted by their noise variances and are summed. The non-homogeneous regions are further classified into point and line regions. Through the use of the average squared gradient, the algorithm is able to extract lines and edges simultaneously within the line regions with the same approach. Edges and lines are simply given by the maxima of the average squared gradient in the direction perpendicular to the edge and line. This direction corresponds to the larger eigenvalue of Γz. To extract lines of a certain width, σ_2 must be suitably chosen. This algorithm is able to extract bar-shaped lines as well as staircase lines. Unfortunately, the squaring operation leads to strongly biased line positions even for lines with only moderately different lateral contrast.

Edge detection approaches are too numerous to mention them all. See, for example, (Dougherty and Bowyer, 1998, Shin et al., 1998) for a performance evaluation of several pixel precise edge detectors. For subpixel edge extraction, one can fit primitives, e.g., planes or extruded 1D $\tanh$ curves as in (Nalwa and Binford, 1986) or an appropriate line or edge model which is selected from the output of a quadrature filter pair (Kisworo et al., 1994), to windows in the image data, which is very computationally expensive. Alternatively, the edge positions can be interpolated to subpixel precision, e.g., from the Laplacian of the image (Huertas and Medioni, 1986) or the second derivative in the direction of the gradient (Busch, 1996, Cumani, 1991). All of the above approaches will often fail to extract edge junctions from the image although they are clearly visible.

2 MODEL-BASED LINE AND EDGE EXTRACTION

2.1 Line and Edge Models

The features we are trying to detect, i.e., edges and lines, can be regarded as one-dimensional manifolds in the two-dimensional space $\mathbb{R}^2$. Both types of features therefore are curvilinear structures $s(t)$ which can be characterized by their typical 1D gray value profile perpendicular to the feature. For edges, we will assume the traditional step edge profile, which is given by

$$f_e(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} . \tag{1}$$

Edges of arbitrary contrast can be obtained by scaling this normalized profile by h, i.e., $hf_e(x)$.

Lines can be regarded as a combination of two step edges. It is useful to consider two different types of profiles, depending on whether the two step edges have the same polarities or not. If the edges have equal polarities (when viewed from the center of the line), we obtain the traditional bar-shaped line profile. Since the contrast on the two sides of the line is usually different, we will use the following profile for lines with equal polarity:

$$f_a(x) = \begin{cases} 0, & x < -w \\ 1, & |x| \leq w \\ a, & x > w \end{cases} , \tag{2}$$

where $a \in [0, 1]$. General lines of height h can be obtained by considering a scaled asymmetrical profile, i.e., $hf_a(x)$. In this definition, w is half the total width of the line. For lines with different polarity we could simply use (2) with $a > 1$. This definition would not be symmetrical, however. Therefore, a line with different polarity is better modeled by

$$f_s(x) = \begin{cases} 0, & x < -w \\ a, & |x| \leq w \\ 1, & x > w \end{cases} , \tag{3}$$

where $a \in [0, 1]$. Again, lines of arbitrary contrast can be obtained by scaling this profile.

2.2 Extraction of Lines with Equal Polarity

The extraction of lines with asymmetrical profiles usually results in biased line positions and widths, as was shown for lines with equal polarity in (Steger, 1998b, Steger, 1998c) and was also considered in the context of edge detection in (Shah et al., 1986, Chen and Medioni, 1989). In (Steger, 1997, Steger, 1998b) it was shown how the bias can be removed from the extracted line positions and widths for lines with equal polarity. We will repeat the most important results here so that we can extend the analysis to the case of lines with different polarity.

In order to detect lines that have the profile (2), we will convolve the image with the derivatives of the Gaussian kernel. This leads to a scale-space description of the line profile:

$$r_a(x, \sigma, w, a) = \phi_\sigma(x + w) + (a - 1)\phi_\sigma(x - w) \tag{4}$$

$$r'_a(x, \sigma, w, a) = g_\sigma(x + w) + (a - 1)g_\sigma(x - w) \tag{5}$$

$$r''_a(x, \sigma, w, a) = g'_\sigma(x + w) + (a - 1)g'_\sigma(x - w) , \tag{6}$$

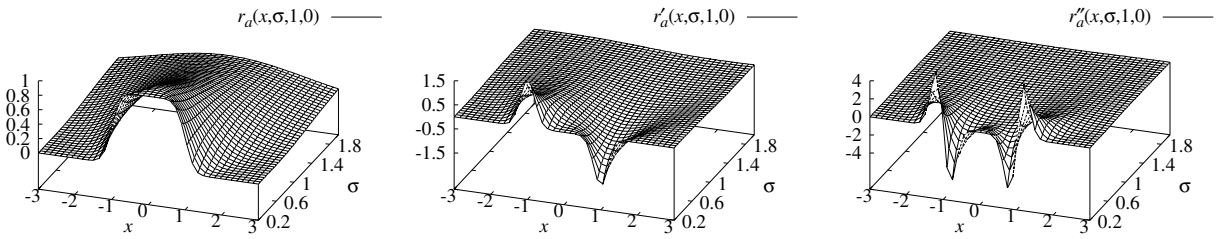


Figure 1: Scale-space behavior of the bar-shaped line f_a with $a = 0$ and $w = 1$ when convolved with the derivatives of Gaussian kernels for $x \in [-3, 3]$ and $\sigma \in [0.2, 2]$.

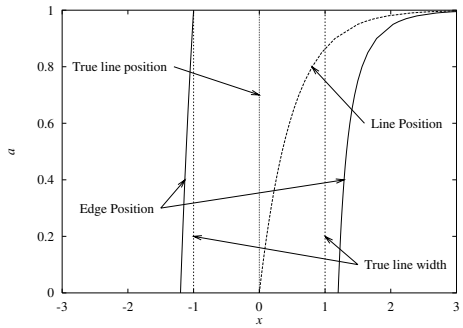


Figure 2: Location of an asymmetrical line and its corresponding edges with width $w = 1$, $\sigma = 1$, and $a \in [0, 1]$.

where $g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{x^2}{2\sigma^2})$ is the Gaussian kernel with standard deviation σ and $\phi_\sigma(x)$ is its integral. The scale-space behavior of a line with $w = 1$ and $a = 0$ is displayed in Figure 1. The line position is given by the maxima (for bright lines) or minima (for dark lines) of $r_a(x, \sigma, w, a)$, i.e., by the points where $r'_a(x, \sigma, w, a) = 0$. We can see that with this definition of the line position we can extract lines with any $\sigma > 0$. However, in (Steger, 1998b) it was shown that for small σ the selection of salient lines will become difficult because the response $r''_a(x, \sigma, w, a)$, on which the selection of salient lines is based, will be small. To make the selection of salient lines easy, $\sigma > w/\sqrt{3}$ should be selected. From (5), we can see that the line position is

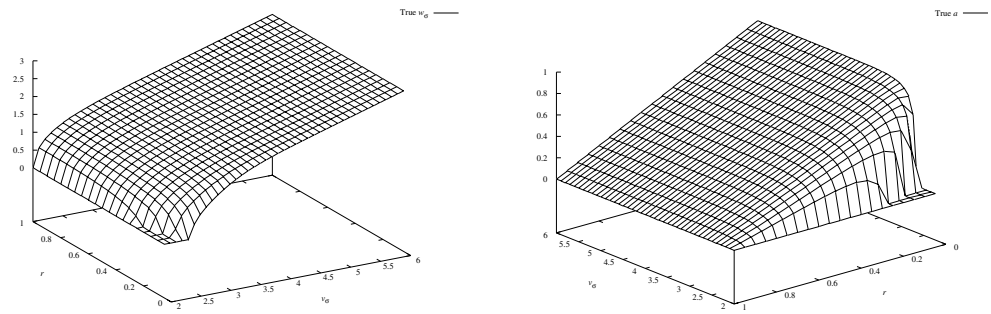
$$l = -\frac{\sigma^2}{2w} \ln(1 - a) \quad . \tag{7}$$

Hence, the line position will be biased if $a \neq 0$.

To extract the line width, we need to recall that a line is a combination of step edges, i.e., a line is bounded by an edge on each side. The edge position can be obtained from the maxima of the absolute value of the first derivative of the smoothed image, i.e., by the maxima of $|r'_a(x, \sigma, w, a)|$ or the zero crossings of $r''_a(x, \sigma, w, a)$, where additionally $r'''_a(x, \sigma, w, a) r'_a(x, \sigma, w, a) < 0$ is required. From the center graph in Figure 1, it can be seen that the edges of the lines, and consequently edges in general, can also be regarded as lines in the image of the first derivative of the input image. In the 2D case, the image of the first derivative corresponds to the gradient image. We will use this interpretation of the edges as lines in the gradient image throughout the rest of the paper.

Unlike the line position (7), the location of the edges can only be computed numerically. Figure 2 gives an example of the line and edge positions for $w = 1$, $\sigma = 1$, and $a \in [0, 1]$. It can be seen that the position of the line and the edges is greatly influenced by line asymmetry. 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.

All these bias effects are not very encouraging if we want to extract lines and their width with high accuracy. Fortunately, the definition of the line position by $r'_a(x, \sigma, w, a) = 0$ and of the edge position by $r''_a(x, \sigma, w, a) = 0$ can be used to model the bias of the line position and width explicitly and hence to remove the bias, as was shown in (Steger, 1998b, Steger, 1998c). One of the properties that makes this analysis feasible is that the biased line positions and widths form a scale-invariant system, which means that we can think in terms of scale-normalized line positions $l_\sigma = l/\sigma$ and line widths $w_\sigma = w/\sigma$. This means that we save one parameter because we can set σ to an arbitrary value, e.g., $\sigma = 1$ to analyze the bias. With this, we can predict two features that we can extract from the image, i.e., the total line width v_σ and the ratio of the gradients r, as a function of the true line width w_σ and the true asymmetry a. More formally, there is a function $f : (w_\sigma, a) \mapsto (v_\sigma, r)$. It can be shown that the inverse f^{-1} of this function exists and is unique. With this, it is possible to remove the bias from the extracted line positions and widths. Figure 3 displays the bias removal function f^{-1}. Note that f^{-1} cannot be described by an analytical formula. It must be computed by a multi-dimensional root finding



(a) True w_σ (b) True a

Figure 3: True values of the line width w_σ (a) and the asymmetry a (b).

algorithm (Press et al., 1992). To make the bias inversion efficient, f^{-1} must be computed once offline and tabulated. Because of the scale-invariance property, the resulting table is an array of only two dimensions (v_σ and r), which makes the table manageable in size.

In 2D, we can model lines as curves $s(t)$ that exhibit a characteristic 1D profile in the direction perpendicular to the line, i.e., perpendicular to $n(t) = s'(t)$. Hence, we can extract lines points in 2D by requiring that the first directional derivative in the direction $n(t)$ should vanish and the second directional derivative should be of large absolute value. The direction $n(t)$ can be obtained for each pixel from the eigenvector corresponding to the eigenvalue of largest magnitude of the Hessian matrix of the smoothed image. The Hessian and the gradient result in a second-degree Taylor polynomial in each pixel, from which we can extract the line position with subpixel accuracy (Steger, 1998b). To extract the line width, the edges on the right and left side of the line are extracted by extracting edge points on a search line of length 2.5σ in the direction $\pm n(t)$. The length of the search line is motivated by the restriction $\sigma > w/\sqrt{3}$. As mentioned above, edges are regarded as bright lines in the gradient image. Therefore, to extract edge points we need the first and second partial derivatives of the gradient image. The gradient image is given by

$$e(x,y) = \sqrt{f_x(x,y)^2 + f_y(x,y)^2} = \sqrt{f_x^2 + f_y^2} \tag{8}$$

where $f(x,y)$ is the image smoothed with $g_\sigma(x,y)$. The partial derivatives are given by:

$$e_x = \frac{f_x f_{xx} + f_y f_{xy}}{e} \tag{9}$$

$$e_y = \frac{f_x f_{xy} + f_y f_{yy}}{e} \tag{10}$$

$$e_{xx} = \frac{f_x f_{xxx} + f_y f_{xxy} + f_{xx}^2 + f_{xy}^2 - e_x^2}{e} \tag{11}$$

$$e_{xy} = \frac{f_x f_{xxy} + f_y f_{xyy} + f_{xx} f_{xy} + f_{xy} f_{yy} - e_x e_y}{e} \tag{12}$$

$$e_{yy} = \frac{f_x f_{xyy} + f_y f_{yyy} + f_{xy}^2 + f_{yy}^2 - e_y^2}{e} \quad . \tag{13}$$

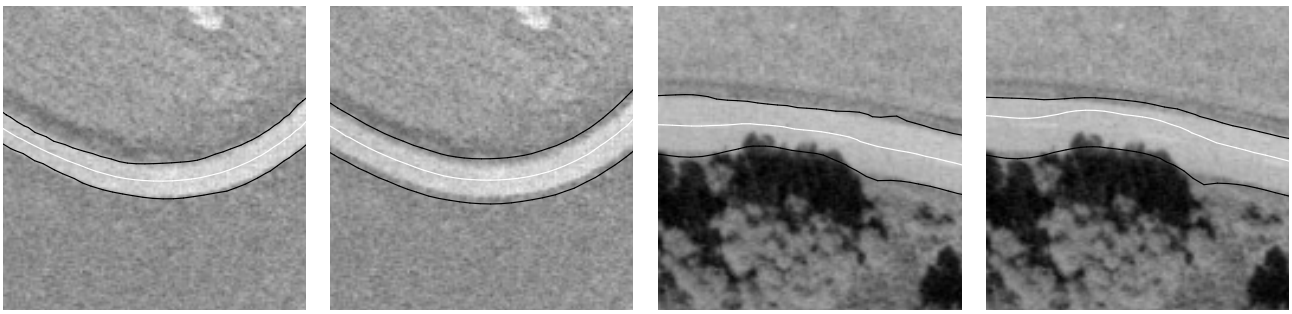
As can be seen, we need the third partial derivatives of the smoothed image, i.e., 8 convolutions in total. For efficiency reasons, the coefficients $e_x, \ldots, e_{yy}$ are computed by convolving the image with 3×3 facet model masks (Steger, 1998b). With this second-degree Taylor polynomial, the edge point extraction is exactly the same as the line point extraction above. Note that in contrast to standard edge detection approaches, the direction perpendicular to the edge is obtained from the Hessian of the gradient image, not from the gradient direction. We will see below what implications this definition has.

The individual line points are linked into lines by an extension of Canny's hysteresis thresholding algorithm (Canny, 1986) which takes the direction of the lines into account and correctly handles junctions (Steger, 1998b).

Figures 4(a) and (c) display the result of extracting lines and their width with bias removal from an aerial image with a reduced resolution of 1 m. To assess the accuracy of the results, they are shown superimposed onto the original image of resolution 0.25 m. For comparison purposes, Figures 4(b) and (d) display the results without bias removal. Evidently, the algorithm was able to correct the line positions and widths successfully with high accuracy.

2.3 Extraction of Lines with Different Polarity

We would now like to use the same scale-space analysis techniques as for lines with equal polarity to design an algorithm that is able to extract lines with different polarity, which returns unbiased line positions and widths. This type of lines



(a) Bias removed (b) Bias not removed (c) Bias removed (d) Bias not removed

Figure 4: Lines and their width detected in an aerial image of a reduced resolution of 1 m. To assess the accuracy of the results, they are shown superimposed onto the original image of resolution 0.25 m.

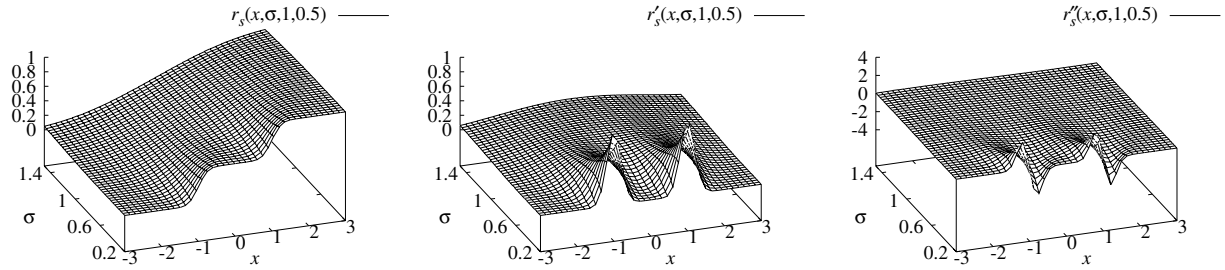


Figure 5: Scale-space behavior of the staircase line f_s with $w = 1$ and $a = 0.5$ when convolved with the derivatives of Gaussian kernels for $x \in [-3, 3]$ and $\sigma \in [0.2, 1.5]$.

does not occur frequently for interesting objects in real images. Nevertheless, lines with different polarity (staircase lines for short) are interesting from a theoretical point of view.

In order to extract line points with the profile given by (3), the profile must be convolved with the derivatives of a Gaussian in order to get a smooth function. The respective responses are given by

$$r_s(x, \sigma, w, a) = a\phi_\sigma(x + w) + (1 - a)\phi_\sigma(x - w) \tag{14}$$

and its derivatives. Figure 5 shows the scale-space behavior of a staircase line with $w = 1$ and $a = 0.5$. As we can see, in the one-dimensional case the position of the lines is given by the inflection point of the smoothed line profile, where additionally the inflection must be "flat," i.e., the line position is given by $r''_s(x, \sigma, w, a) = 0$ with $r'''_s(x, \sigma, w, a)r'_s(x, \sigma, w, a) > 0$. This is very similar to the definition of edges in the 1D case given in Section 2.2, where the inflection point is required to be non-flat. Considering Figure 5, it is clear that the flat inflection points will turn non-flat if σ is chosen large enough compared to the line width w, i.e., the line will turn into an edge. Because for $a = 0.5$ the line or edge position is at $x = 0$ for all σ, this can be analyzed analytically by requiring $r'''_s(0, \sigma, w, 0.5) = 0$. With this, by a straightforward calculation it can be shown that lines can only be extracted for $\sigma < w$. For other values of a, it can be expected that the line can be extracted for even smaller values of σ only. Since for applications σ is constant, while a and w vary in the image, it is interesting to determine the range of a and w for which lines can be extracted with a given σ. It can be shown (Steger, 1998c) that lines can only be extracted for

$$1 - a_s(w, \sigma) < a < a_s(w, \sigma) \ , \tag{15}$$

where

$$a_s(w, \sigma) = \frac{1}{1 - \dfrac{w^2 - \sigma^2 + w\sqrt{w^2 - \sigma^2}}{w^2 - \sigma^2 - w\sqrt{w^2 - \sigma^2}} e^{-2\frac{w\sqrt{w^2-\sigma^2}}{\sigma^2}}} \ . \tag{16}$$

Figure 6 shows the range of w and a for which lines can be extracted with $\sigma = 1$. We can see that the range of asymmetries grows for larger line widths. Furthermore, theoretically it is possible to extract lines of arbitrary width with $\sigma = 1$ in the continuous case.

Finally, it should be noted that salient staircase lines must be selected based on the magnitude of r'''_s since the first two derivatives will be very small and zero, respectively, at line points. As was the case for the bar-shaped line in Section 2.2, the magnitude of r'''_s does not exhibit a unique maximum if σ is small compared to w. Since salient lines are selected

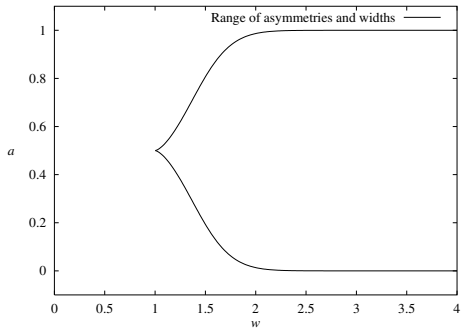
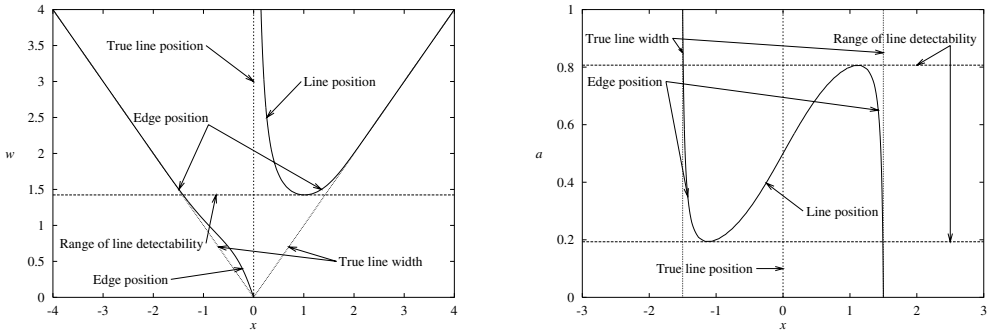


Figure 6: Range of widths w and asymmetries a for which staircase lines can be extracted with $\sigma = 1$. The extractable range is contained between the two curves.



(a) (b)

Figure 7: (a) Location of a staircase line with $a = 0.75$, $w \in [0, 4]$ and its edges for $\sigma = 1$. (b) Location of a staircase line with $w = 1.5$, $a \in [0, 1]$ and its edges for $\sigma = 1$.

based on this value it is necessary to derive a restriction on σ similar to the restriction $\sigma > w/\sqrt{3}$ for bar-shaped lines. This analysis can only be carried out for symmetrical staircase lines with $a = 0.5$ since only then the line position can be determined analytically. With these parameters, we obtain

$$\sigma \geq w\sqrt{\frac{3-\sqrt{6}}{3}} \quad . \tag{17}$$

Combined with the above restriction $\sigma < w$, the value of σ should be chosen in the range

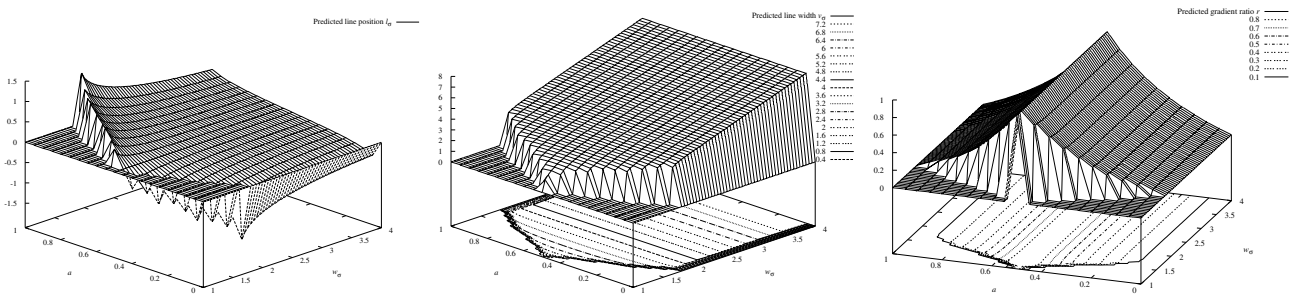
$$w\sqrt{(3-\sqrt{6})/3} \leq \sigma < w \tag{18}$$

if lines of width w are to be extracted.

If staircase lines are extracted with the above definition and their edges with the definition of Section 2.2, the results will again generally be biased. Figure 7(a) shows the line and edge positions for a staircase line with $a = 0.75$ for $w \in [0, 4]$ extracted with $\sigma = 1$. As can be seen, lines can only be extracted for $w > 1.424$ for this choice of a. Furthermore, the line position moves to the weak side of the line, i.e., the side with the smaller gradient, for small w. For large w the extracted line widths correspond closely to the true line widths, while the extracted line position asymptotically approaches the true line position. However, this convergence is very slow. It is also instructive to examine the behavior of the line positions and widths if σ and w are kept constant, while the asymmetry a is varied. An example of this behavior for $\sigma = 1$ and $w = 1.5$ is displayed in Figure 7(b). As is to be expected, the behavior of the line and edge position is symmetrical with respect to $a = 0.5$. The line position is biased by a large amount for small and large a. The edge position is also biased, albeit by a smaller amount, for these choices of a.

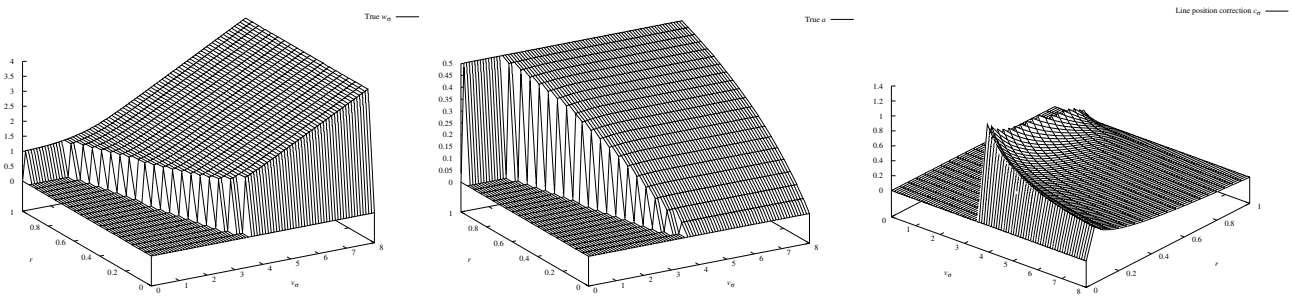
From these examples it can be seen that the extracted line and edge positions of staircase lines generally exhibit a bias, similar to the bar-shaped lines. The major difference is that for staircase lines the extracted line position always remains within the true line. Furthermore, the extracted line width is always too small, while it is always too large for bar-shaped lines. Therefore, for staircase lines a bias removal algorithm is also essential to obtain meaningful results.

In order to model the bias of lines with different polarity the technique of Section 2.2 can be used. As was the case for bar-shaped lines, the line positions and widths of staircase lines form a scale-invariant system. Therefore, we can work



(a) (b) (c)

Figure 8: (a) Predicted line position l_σ of the staircase line f_s for $w_\sigma \in [1,4]$ and $a \in [0,1]$. (b) Predicted line width. (c) Predicted gradient ratio r.



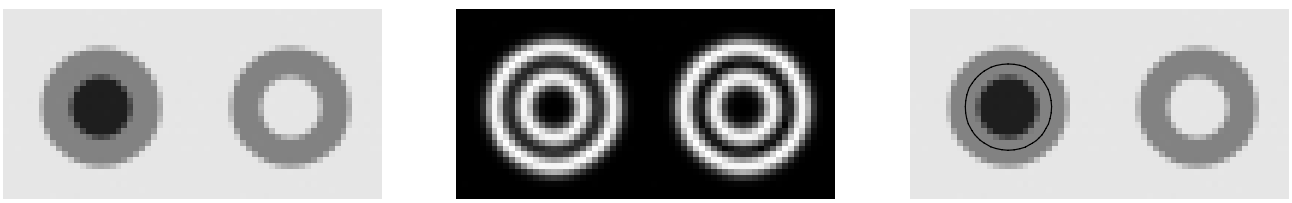
(a) (b) (c)

Figure 9: (a) True values of the line width w_σ of the staircase line f_s. (b) True values of the asymmetry a. (c) Line position correction. Note that (c) is rotated by $180°$ with respect to (a) and (b).

with scale-normalized quantities w_σ and v_σ as above. Thus, the predicted line position l_σ can be calculated for all w_σ and $a \in [0,1]$. The result is shown in Figure 8(a). We can see that the bias of the line position is largest for small line widths and large asymmetries. Furthermore, the bias is symmetrical with respect to $a = 0.5$. Note that the positions are only defined within the range of a and w defined by (15). Furthermore, the predicted line width w_σ and gradient ratio r can be calculated. They are displayed in Figures 8(b) and (c) along with their contour lines.

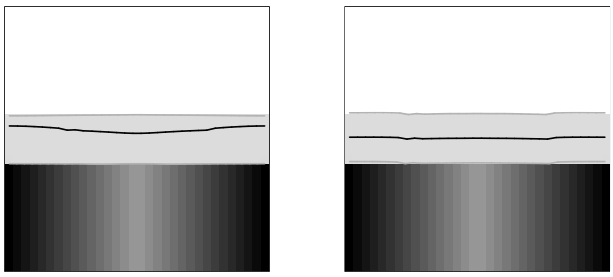
As is obvious from Figures 8(b) and (c), for $a \leq 0.5$ the bias function is invertible since the contour lines of v_σ and r intersect almost perpendicularly everywhere. Hence, the inverted bias function can be calculated by a multi-dimensional root finding algorithm to obtain the true values of w_σ and a. They are shown in Figures 9(a) and (b). Furthermore, the absolute value of the correction c_σ to be applied to the line position in order to obtain the true line position is displayed in Figure 9(c). The only remaining question is how the extraction algorithm should decide whether the true asymmetry is smaller or larger than 0.5. This can simply be done based on the gradient directions at one of the two edge points and on the direction perpendicular to the line. If the dot product of the two vectors is larger than zero, the bright side of the line is the right side, and vice versa. If the bright side of the line and the side with the weaker gradient are equal, the asymmetry must be chosen larger than 0.5, otherwise smaller.

The discussion so far has been concerned with the extraction of line points in 1D. It is not immediately obvious how to derive a 2D extraction algorithm for lines with different polarity. However, if the gradient image is considered, it can be seen that such a line manifests itself as a dark line there. An example of this is displayed in Figure 10(a), where two rings of lines of different and equal polarity are displayed. In the gradient image in Figure 10(b) the line of different polarity



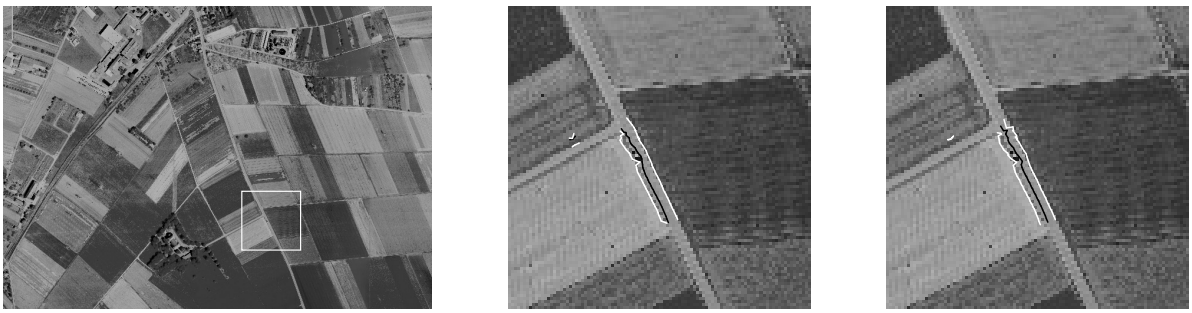
(a) Input image (b) Gradient image (c) Extracted line

Figure 10: (a) An image containing two rings of lines with different and equal polarities, respectively. (b) The gradient image shows that both types of lines result in dark lines in the gradient image. (c) Extracted staircase lines.



(a) Bias not removed (b) Bias removed

Figure 11: Result of extracting lines and their width from a test image containing a line with varying asymmetry (a) without and (b) with bias removal.



(a) Aerial image (b) Bias not removed (c) Bias removed

Figure 12: Aerial image (a) containing a line with different polarity in the part marked by the square. Lines extracted from the marked part in (a) without (c) and with (d) bias removal.

can be seen as a dark line. Therefore, one way to extract lines with different polarity is to use the facet model line detector used to extract the line width in Section 2.2 to extract dark lines from the gradient image. However, as Figure 10(b) shows, lines with equal polarity also result in dark lines in the gradient image. Since only lines with different polarity should be detected, this simple approach is not sufficient.

Fortunately, there is a simple criterion by which lines of different and equal polarity can be discerned. From (14) it is obvious that the gradients point in the same direction on both sides of a line of different polarity, while they point in opposite directions for lines with equal polarity. Thus, to discern lines of different and equal polarity, the gradients at the edge positions in both directions perpendicular to the dark line in the gradient image are evaluated, and their dot product is calculated. Only if the dot product is positive, a line of different polarity has been found. However, it is useful to be more restrictive in the selection of lines. For example, junctions of three edges are also extracted as dark lines from the gradient image in the vicinity of the junction. Therefore, in the implementation a dark line point is only accepted as a line point of different polarity if the directions of the gradients are sufficiently close. Here, sufficiently close is defined by the gradient directions forming an angle of less than $30°$. With these criteria, lines points of different polarity can be extracted fairly reliably.

To link the individual line points into lines the same linking algorithm as for bar-shaped lines is used (Steger, 1998b, Steger, 1998c). The two hysteresis thresholds are applied to the eigenvalues of the dark lines in the gradient image, which are equivalent to the third directional derivative in the direction perpendicular to the original line. Figure 10(c) displays the results of extracting lines of different polarity with $\sigma = 1.5$ from the image in Figure 10(a). As can be seen, with the approach discussed above the line of equal polarity is not extracted.

Figure 11(a) exhibits the results of extracting the line position and width with $\sigma = 1.7$ from a test image containing a line with varying asymmetry. As can be expected, the line position is strongly biased. In contrast, the line width corresponds to the true line width for most of the line. The result of applying the bias correction to the extraction results is shown in Figure 11(b). As can be seen, the extracted line position corresponds to the true line position, while the extracted line width is very close to the true line width.

To conclude this section, Figure 12(a) displays an aerial image containing a line of different polarity in the part marked by the square. This image was obtained from (Fritsch and Sester, 1994). The results of extracting lines of different polarity with $\sigma = 1.4$ and thresholds of 2 and 7 without and with applying the bias removal are displayed in Figures 12(b) and (c), respectively. As can be seen, only the road corresponding to the staircase line has been extracted. Since the line has a fairly

large asymmetry, the uncorrected line positions and widths are severely biased. This bias is removed by the correction step, as is clear from Figure 12(c). The extracted line corresponds closely to the true position and width of the road.

As noted above, the algorithm proposed in this section makes sure not to extract lines of equal polarity. This was explicitly done because the line position and width correction is completely different for the two different types of lines. In some cases it might be useful to extract lines of equal and different polarity at the same time. The easiest way to achieve this would be to regard lines of equal polarity as dark lines in the gradient image also. Note that the modeling of the bias is not affected by this, since it only depends on the width and asymmetry of the line. In this case, however, the line position would not be given by (7), and hence the position correction would be invalid. Instead, an explicit correction would have to be computed numerically in the same manner as for lines of different polarity.

2.4 Edge Extraction

As discussed in Section 2.2, in a gradient image edges appear as bright lines. Hence, with the line extraction and linking algorithm of Section 2.2 it is possible to design an algorithm that turns an arbitrary gradient-based edge detector, such as the Canny (Canny, 1986), (isotropic) Deriche (Deriche, 1987, Lanser and Eckstein, 1992), (isotropic) Shen (Shen and Castan, 1992, Lanser and Eckstein, 1992) or even the Sobel operator, into a subpixel edge detector.

Before the edge detection algorithm will be described, a few words on the model of the underlying edge are necessary. Most approaches use the step edge model (Canny, 1986, Deriche, 1987, Lanser and Eckstein, 1992, Shen and Castan, 1992), i.e., assume the edge of height h to be given by $hf_e(x)$, where $f_e(x)$ is given by (1). Other approaches assume the edge to be a Gaussian-blurred step edge, i.e., its profile to be given by $h\phi_\sigma(x)$ (Lindeberg, 1998). While both these assumptions are useful, since they simplify the analysis of the operator considerably, for our purposes they are not strictly necessary. The main point about these step edge profiles is that they are symmetric with respect to the inflection point at $x = 0$. Therefore, if the absolute value of the first derivative of these profiles are taken, symmetrical line profiles are obtained. It is obvious that under this condition, and the condition that the smoothing filter the edge operator uses is symmetrical with respect to the origin, isolated lines, and hence also edges, will be extracted in the correct subpixel position. Of course, if the edge profile is not symmetric to its inflection points, the edge locations will be biased (Ulupinar and Medioni, 1990). If the correct edge model can be derived analytically, it is possible to model this bias with the techniques introduced in Sections 2.2 and 2.3, and hence to remove the bias.

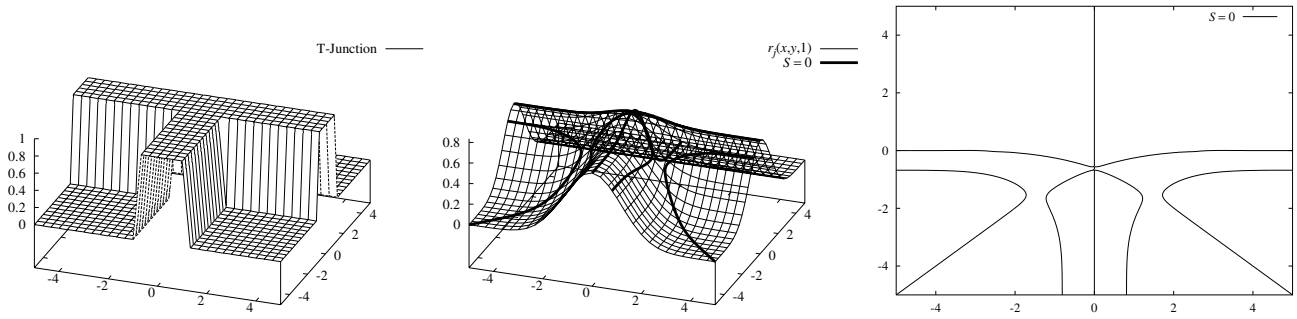
As a result of the above discussion, the facet model edge detection scheme used in Section 2.2 to extract the line width and in Section 2.3 to extract the line positions (as dark lines in the gradient image) and widths can be used to detect single edge points, and that the linking algorithm can be used almost unaltered. Only two modifications need to be made to the linking algorithm. The first one is concerned with the thresholds used in the linking algorithm. To obtain the standard feature used for edge selection — the gradient magnitude — the responses of the line detector are substituted by the response of the edge operator at the corresponding point. The second, and most important, change is how to implement the algorithm efficiently. It is obvious that the facet model line detector does not need to be run on the entire image since the lower threshold of the linking algorithm determines the points which cannot be edge points. Hence, the line detector only needs to be applied to the region of the image in which the response of the edge detector is higher than the lower hysteresis threshold. With these modifications, the resulting subpixel edge detector is only about 15% slower than the typical edge detection cycle of running the edge operator, doing a combined non-maximum suppression and hysteresis threshold operation, thresholding the resulting image, calculating a skeleton from this region, and linking the edge points into contours. This is a very small price to pay for the added bonus of subpixel accurate results. Examples of the results obtainable with the proposed edge detector are given in the next section.

3 MISSING JUNCTIONS IN LINE AND EDGE EXTRACTION

Missing junctions in edge extraction are a well-known problem (Rothwell et al., 1994). Unfortunately, the also occur for the line and edge extraction algorithms proposed in this paper. In this section, we will use explicit models to get a qualitative impression of the behavior of the line and edge detectors proposed above in the vicinity of junctions and derive an algorithm that is able to extract the complete junction information from an image.

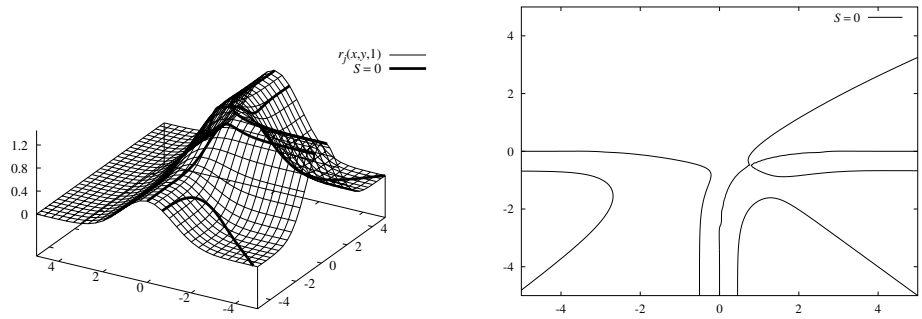
We will start by modeling junctions of lines. Consider a T-junction where lines of different width and different contrast meet. A model for this type of junction is

$$f_j(x,y) = \begin{cases} h_1, & x \geq 0 \wedge |y| \leq w_1 \\ h_2, & x < 0 \wedge |y| \leq w_2 \\ h_3, & |x| \leq w_3 \wedge y < -w_1 \\ 0, & \text{otherwise} \end{cases}. \tag{19}$$



(a) T-junction (b) Extracted lines (c) Lines in the (x, y)-plane

Figure 13: (a) A T-junction with lines of equal width and contrast. Extracted lines by the De Saint-Venant condition for a T-junction with lines of equal width and contrast: (b) mapped onto the response $r_j(x, y, 1)$, and (c) in the (x, y)-plane.



(a) Extracted lines (b) Lines in the (x, y)-plane

Figure 14: Extracted lines by the De Saint-Venant condition for a T-junction with lines of equal width and different contrasts of $h_1 = h_3 = 2$ and $h_2 = 1$: (a) mapped onto the response $r_j(x, y, 1)$, and (b) in the (x, y)-plane.

The scale-space description of this junction is given by

$$\begin{aligned} r_j(x, y, \sigma) &= h_1(\phi_\sigma(y + w_1) - \phi_\sigma(y - w_1))\phi_\sigma(x) + \\ &\quad h_2(\phi_\sigma(y + w_2) - \phi_\sigma(y - w_2))(1 - \phi_\sigma(x)) + \\ &\quad h_3(\phi_\sigma(x + w_3) - \phi_\sigma(x - w_3))(1 - \phi_\sigma(y + w_1)) \end{aligned} \tag{20}$$

and its partial derivatives of first and second order. Figures 13(a) and (b) show a T-junction with lines of equal widths $w_1 = w_2 = w_3 = 1$ and equal contrasts $h_1 = h_2 = h_3 = 1$, and the corresponding response $r_j(x, y, 1)$ for $(x, y) \in [-5, 5] \times [-5, 5]$.

In order to analyze the scale-space behavior of the line positions the zero crossings of the first directional derivative in the direction of the maximum second directional derivative must be obtained. To analyze the behavior of lines in the 2D case, we can use De Saint-Venant's condition $S = 0$ (Koenderink and van Doorn, 1994). This condition is given by

$$S = \frac{\nabla r_j^T H \nabla r_j^\perp}{\nabla r_j^T \nabla r_j} = \frac{r_x r_y (r_{xx} - r_{yy}) - (r_x^2 - r_y^2) r_{xy}}{r_x^2 + r_y^2} = 0 \ . \tag{21}$$

This condition will extract a superset of the lines the proposed algorithm extracts because it will also extract the points where there is a restricted extremum in the direction of minimum second directional derivative, and because it does not discern between restricted maxima and minima. Nevertheless, because the extracted lines form a superset of the lines returned by the proposed algorithm, the reasons for missed junctions can be studied.

Figures 13(b) and (c) show the lines extracted by the condition $S = 0$ for the T-junction of Figure 13(a). One thing immediately strikes the eye: the "hump" in the middle of the Gaussian smoothed junction. As can be seen, the three extracted lines seem to intersect in the middle of the junction, at the maximum of the hump. However, closer inspection reveals that the line of the y-axis actually is a restricted maximum in the direction of the *minimum* second directional derivative in the vicinity of the junction. Therefore, failures to extract junctions, even for lines with identical contrast can be explained. From this it might seem that one should allow lines to be restricted extrema in the direction of the minimum second directional derivative as well, in order to extract all junctions. The next example shows that this is not the case.

Figure 14 shows a junction with $h_1 = h_3 = 2$ and $h_2 = 1$, i.e., the left line on the y-axis is now darker than the other two lines. As can be seen, the line on the left part of the y-axis again changes to a restricted maximum in the direction of the

minimum second directional derivative. However, as this line approaches the other two lines it rapidly turns away from the junction, and ends up running parallel to the line on the y-axis. Therefore, allowing lines to be restricted extrema in the direction of the minimum second directional derivative is definitely a wrong approach. What is striking, though, is that the direction of the line on the y-axis at the point where it changes its type points exactly at the junction. More examples with different configurations seem to indicate that this is true in general (Steger, 1998c). Therefore, a good algorithm to extract the missed junctions is to search for other lines along the last extracted line direction. As can be seen from the examples, the lines not continuing to the junction will in general be the lines with the smaller contrast. Hence, while following the line, the gray value in the image should increase monotonically for bright lines and decrease monotonically for dark lines, respectively. Since it is desirable to avoid using the image intensity directly, this condition can be reformulated to the condition that the dot product of the gradient of the image and the line direction must always be positive for bright lines and always negative for dark lines. Of course, the length of the line along which the search is performed should not be arbitrarily long. A good restriction is to use the same length 2.5σ as for the determination of the line width.

Let us now analyze the behavior of edges in the vicinity of junctions. Consider a T-junction where three areas of different intensity meet. A suitable model for this type of junction is

$$f_j(x,y) = \begin{cases} 0, & y < 0 \\ 1, & y \geq 0 \wedge x < 0 \\ h, & y \geq 0 \wedge x \geq 0 \end{cases} . \tag{22}$$

It is obvious that this models any possible configuration of intensities, since arbitrary intensities can always be scaled and translated such that one is zero, while another one is one. Hence the only free parameter is the third intensity. For a Gaussian smoothing filter, the scale-space description of this junction is given by

$$r_j(x,y,\sigma) = \phi_\sigma(y)((h-1)\phi_\sigma(x)+1) \tag{23}$$

and its partial derivatives of first and second order.

As described in Section 2.2, the edge points in the one-dimensional case are given by the maxima of $|r'_e(x,\sigma)|$, or equivalently by the zero crossings of $r''_e(x,\sigma)$, where additionally $r'''_e(x,\sigma)r'_e(x,\sigma) < 0$ is required. As we have seen, the first interpretation quite naturally leads to the conclusion that edges can be modeled as lines in the gradient image. In the two-dimensional case, these two edge definitions have been translated to the requirement that an edge is either given by the maxima of the first directional derivative in the direction of the gradient (Haralick, 1984) or by the zero crossings of the Laplacian (Marr and Hildreth, 1980). The first condition can be expressed as

$$Dr_e = \frac{r_x^2 r_{xx} + 2r_x r_y r_{xy} + r_y^2 r_{yy}}{r_x^2 + r_y^2} = 0 , \tag{24}$$

while the second condition can be expressed as

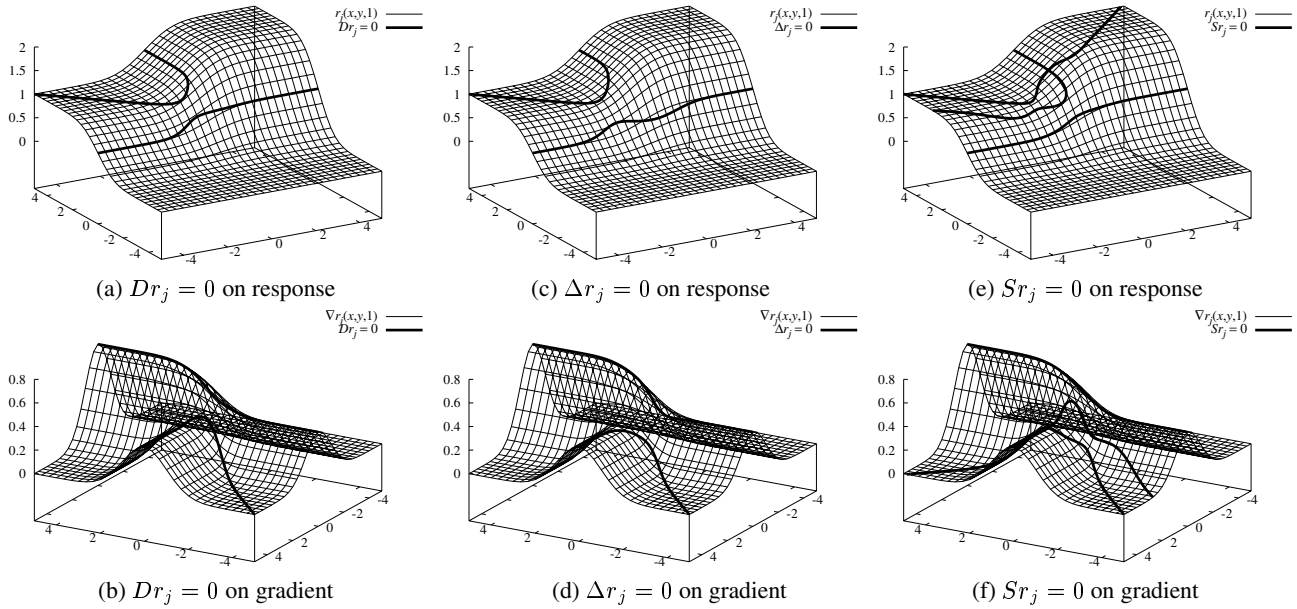
$$\Delta r_e = r_{xx} + r_{yy} = 0 . \tag{25}$$

As is well known, these two definitions are equivalent if and only if the mean curvature of the image vanishes (Berzins, 1984). This difference is especially noticeable for corners, where the first definition will move the edge positions to the inside of the corner, while the second definition will broaden the corner. In contrast, the interpretation of edges as lines in the gradient image leads again to the use of the De Saint-Venant condition to model the behavior of edges in 2D. In this case, edges are given by

$$Sr_e = \frac{e_x e_y (e_{xx} - e_{yy}) - (e_x^2 - e_y^2)e_{xy}}{e_x^2 + e_y^2} = 0 , \tag{26}$$

where $e_x, \ldots, e_{yy}$ are calculated according to (9)–(13). We will use these three definitions to explain the often counterintuitive failures of edge detectors to extract junctions.

In order to analyze the behavior of the different edge definitions at junctions, the response r_j can be substituted into (24)–(26). Figure 15 shows the edge positions extracted with the different edge definitions for a junction with $h = 2$ mapped onto the response r_j and onto the absolute value of the gradient $\|\nabla r_j\|$. More examples can be found in (Steger, 1998c). As was the case for the analysis of the line junctions, the spurious edges for the three definitions have to be disregarded. For $Dr_j = 0$ these are the points where the absolute value of the gradient has a minimum in the direction of the gradient. For $\Delta r_j = 0$ they are the flat inflection points, while for $Sr_j = 0$ they are the points that are restricted minima in the direction of the maximum second directional derivative and the minima and maxima in the direction of the minimum second directional derivative. If this is kept in mind, the first interesting fact noticeable from these figures is that the edge positions extracted by $Sr_j = 0$ approach the junction much closer than the edges extracted by the other two definitions. Therefore, it can expected that the first two definitions will miss junctions much more frequently than the



(a) $Dr_j = 0$ on response (c) $\Delta r_j = 0$ on response (e) $Sr_j = 0$ on response

(b) $Dr_j = 0$ on gradient (d) $\Delta r_j = 0$ on gradient (f) $Sr_j = 0$ on gradient

Figure 15: Extracted edges for a T-junction with $h = 2$ by $Dr_j = 0$ mapped onto r_j (a) and onto $\|\nabla r_j\|$ (b); by $\Delta r_j = 0$, (c) and (d); and by $Sr_j = 0$, (e) and (f). The graphs in the lower row are rotated by $120°$ with respect to the graphs in the upper row.

third definition, since for this definition the extracted edge often will approach the junction to within one pixel, so that the linking algorithm will be able to extract a junction. Furthermore, the direction of the edges extracted by the first two definitions in the vicinity of the junction are much worse than by the third definition. As is the case for lines, the edges extracted by $Sr_j = 0$ will point directly at the junction, whereas for the other two definitions this is not necessarily true, a fact that was also noted in (de Micheli et al., 1989). It is a surprising fact that the edge directions for $Sr_j = 0$ are much more stable than the gradient-based directions, since the former ones are based on the second derivatives, while the latter ones are based only on first derivatives.

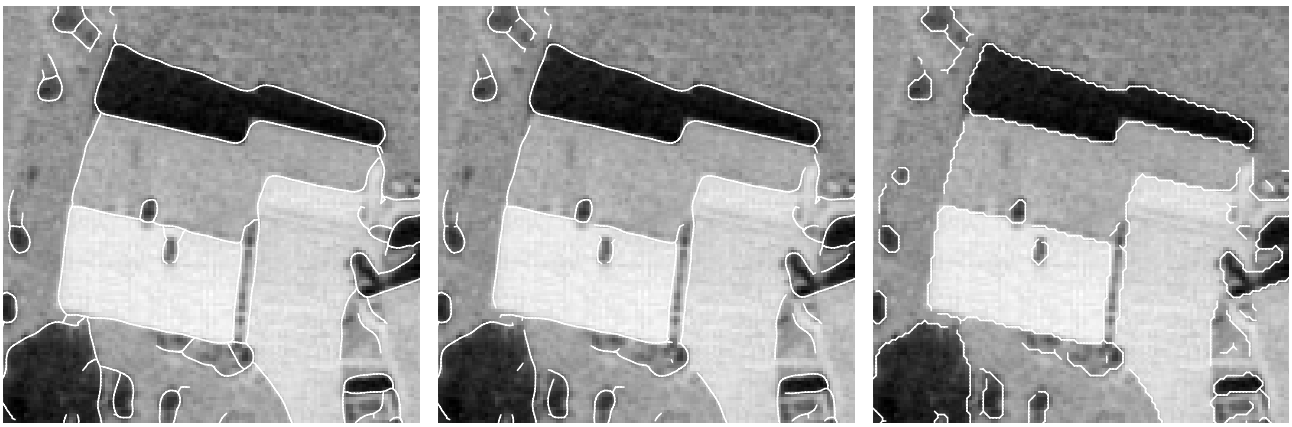
This discussion gives another good reason to regard edges as bright lines in the gradient image. While the edge positions are equally good outside of junction areas, the line-based approach has a clear advantage in junction areas since the edge directions still point in the right direction in the vicinity of a junction for the edge that cannot join the junction. Therefore, the algorithm proposed above for lines can be used to complete the missing junctions for edges as well.

Let us conclude this section by giving an example of the proposed approach to complete missing junctions for edges. More examples for lines and edges can be found in (Steger, 1998c). Figure 16 displays the results of three different extraction schemes to illustrate the results of the analysis of the behavior of the different edge definitions at junctions. In Figure 16(c), the results of the edge detection with pixel resolution using the definition in (24) are shown, i.e., the edges were computed by a Gaussian gradient operator, non-maximum suppression in the direction of the gradient, a hysteresis thresholding operation, computing the skeleton of the resulting region, and linking the edge points into contours. As was to be expected by the analysis above, this algorithm fails to detect almost all of the salient junctions of the building. If the edges are extracted to subpixel resolution by the definition in (26), some of the junctions are detected because the more robust estimation of the edge direction allows the edge to come within one pixel of the junction, as is shown in Figure 16(b). From this figure, also the accuracy of the extracted edge positions becomes readily observable. Especially the gable and perimeter edges of the roof are almost perfect straight lines. Finally, in Figure 16(a), the results of edge extraction with the additional step of completing the junctions is shown. It is apparent that the algorithm was successful in extracting all of the junctions in this part of the image, and has therefore produced a complete description of the house roof.

4 LINE AND EDGE EXTRACTION IN COLOR IMAGES

Sometimes it is useful to extract lines and edges in color images. More specifically, this is true if we cannot find a color-space transformation that gives us a gray value image in which the lines or edges have sufficient contrast. Therefore, in this section the subpixel edge and line extraction algorithms proposed in this paper will be extended to multispectral images of any number of channels.

We will start by extending the edge detection algorithm. As described above, edges can be regarded as bright lines in the gradient magnitude image. Thus, to make use of the subpixel edge detector we need an equivalent of the gradient



(a) Extracted edges (b) Junctions not completed (c) Edges at pixel resolution

Figure 16: Edges detected in an aerial image of resolution 0.25 m. (a) Edges extracted with the proposed algorithm. (b) Edges extracted without junction completion. (c) Edges extracted with pixel resolution by $Dr_e = 0$.

magnitude. Let us regard the multispectral image as a function $f : \mathbb{R}^2 \mapsto \mathbb{R}^n$, where n is the number of channels in the image. For multivalued functions, there is a natural extension of the gradient: the metric tensor G, which gives us for every vector $v \in \mathbb{R}^2$ the rate of change of f in the direction v. For notational convenience, we will regard G as a matrix. Thus, the rate of change of the function f in the direction v is given by $v^T G v$, where (Di Zenzo, 1986, Cumani, 1991)

$$G = \begin{pmatrix} f_x^T f_x & f_x^T f_y \\ f_x^T f_y & f_y^T f_y \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n \frac{\partial f_i}{\partial x}\frac{\partial f_i}{\partial x} & \sum_{i=1}^n \frac{\partial f_i}{\partial x}\frac{\partial f_i}{\partial y} \\ \sum_{i=1}^n \frac{\partial f_i}{\partial x}\frac{\partial f_i}{\partial y} & \sum_{i=1}^n \frac{\partial f_i}{\partial y}\frac{\partial f_i}{\partial y} \end{pmatrix} . \tag{27}$$

Of course, for edge extraction the partial derivatives are computed by convolving the image with the appropriate derivatives of Gaussian masks. Some approaches (Sochen et al., 1998) add the identity matrix to G. This added term will not cause major differences, especially in noisy images, where it is hardly discernible from noise. As in the 1D case, we define the gradient direction by the vector v in which the rate of change is maximum. The vector v can be obtained by computing the eigenvectors of G. The largest eigenvalue of G is the equivalent of the gradient magnitude, i.e., the maximum rate of change, in the 1D case. Contrary to the 1D case, the rate of change perpendicular to v is usually not 0. For $n = 1$, the above definition is equal to the squared gradient magnitude. The only difference to the usual gradient is that we can only determine the orientation of the edge within the interval $[0 : \pi)$ from G. But since we regard edges as lines in the gradient image, this information is unnecessary anyway.

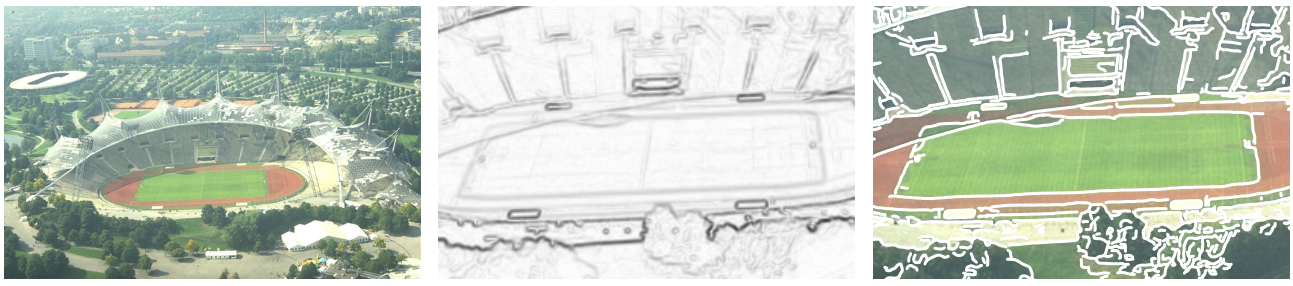
Figure 17 compares the results of extracting edges from a color image of the Olympic Stadium in Munich with the results of extracting edges from an equivalent gray value image. As we can see from Figure 17(b) the borders of the soccer field are clearly visible in the color gradient image. In contrast, in the corresponding gray value gradient image the borders cannot be distinguished. Consequently, the borders of the soccer field have been extracted from the color image (Figure 17(c)), while they have been missed in the gray value image (Figure 17(f)).

To extend the line extraction algorithm to color images, we can regard lines as dark lines in the color gradient magnitude image, as described in Section 2.3. Note that this is reasonable because in a multispectral image lines may have a bright bar-shaped profile in one channel, a dark bar-shaped profile in another channel, and a staircase profile in a third channel. Therefore, the distinction between these types of lines is not meaningful in color images. With this definition of lines, the color line detector is straightforward to implement. Unfortunately, what cannot be implemented is the line position and width correction since, as mentioned above, the different channels may have different line profiles, which have different biases, but are combined into a single image by (27). Therefore, line position and width correction in multispectral images, unfortunately, seems to be impossible.

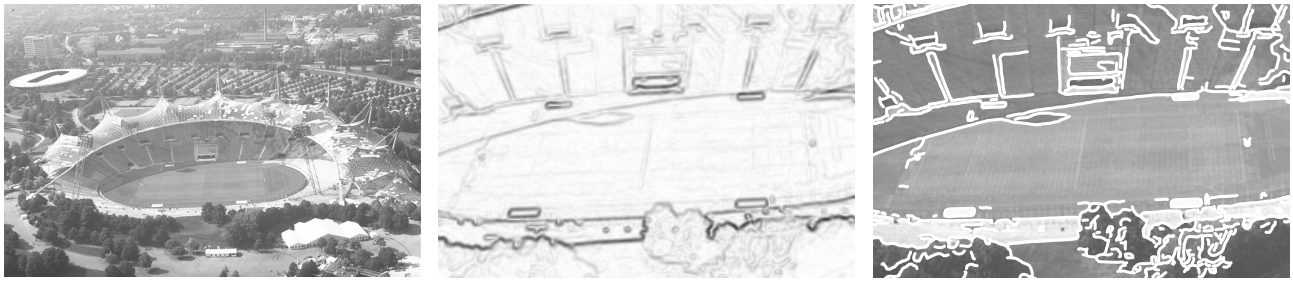
Figure 18 displays the lines extracted from a color image of a set of cables with the proposed approach with $\sigma = 3.5$. Note that the line positions and widths are very good for most of the image. Only the bottom line shows a significant bias.

5 CONCLUSIONS

In this paper, the subpixel accurate line extraction algorithm proposed in (Steger, 1998b), which extracts bright or dark lines lines with different polarity, has been extended to handle lines with equal polarity. A scale-space analysis similar to the analysis in (Steger, 1998b) has been carried out to explicitly model the bias of the line positions and widths of lines

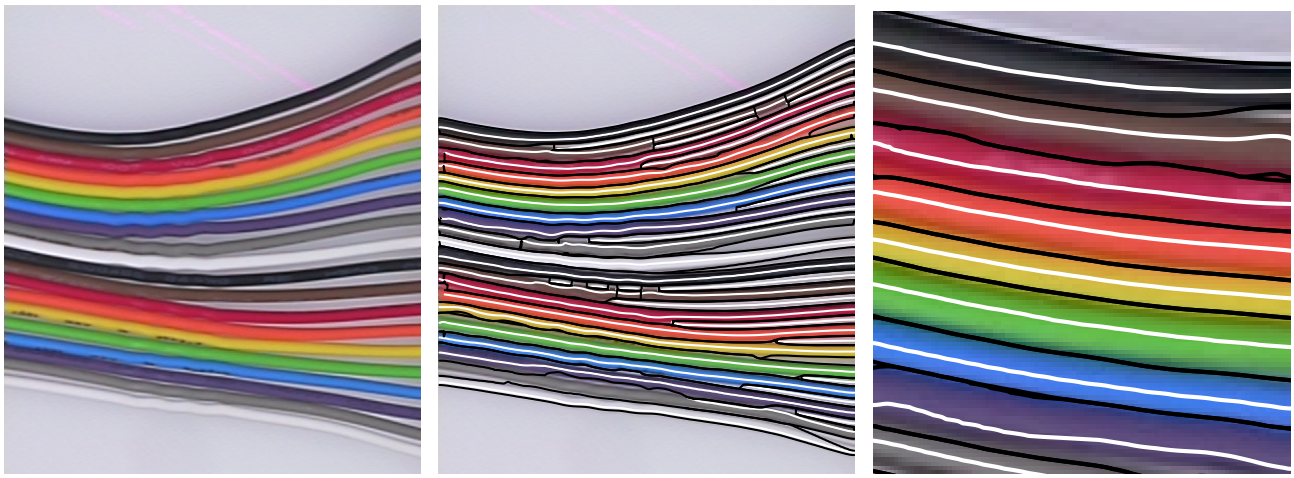


(a) Color image (b) Color gradient magnitude (c) Color edges



(d) Gray image (e) Gradient magnitude (f) Gray value edges

Figure 17: Comparison between color and gray value edge detection. (a) Color image of the Olympic Stadium in Munich. (b) Color gradient magnitude in the central part of (a); high gradients are displayed dark. (c) Extracted color edges. (d) Gray value image corresponding to (a). (e) Gray value gradient magnitude. (f) Extracted gray value edges.



(a) Color image (b) Extracted lines (f) Detailed view

Figure 18: (a) Color image of cables. (b) Extracted color lines; lines are shown in white, while their width is shown in black. (c) Detailed view of the left upper part of (b).

with equal polarity. With this, it is shown that the line position is generally biased if the line exhibits different lateral contrast, while the line width is always biased. From the scale-space analysis an efficient method to remove the bias from the line position and width is derived. Additionally, a subpixel accurate edge extraction algorithm is proposed by regarding edges as bright lines in the gradient image. A performance analysis (Steger, 1998c, Steger, 1998a) shows that edges and lines can be extracted with better than 1/25 pixel accuracy under good conditions.

The behavior of the line and edge detectors in junction areas is analyzed. For the edge detector, the two traditional definitions as zero crossings of the second derivative in the direction of the gradient and the zero crossings of the Laplacian are also considered. This analysis shows the reasons why junctions are often extracted incompletely. From this analysis, a new and efficient method to complete missing junctions is derived.

The proposed line and edge extraction algorithms have been extended to handle multispectral images. Unfortunately, the line position and width correction seems to be an elusive goal for multispectral lines. Nevertheless, for lines with low asymmetry in all bands, the line position and width will be fairly accurate.

A goal for future research is to combine line and edge extraction into a single operator. This is useful for applications that require lines as well as edges as low-level primitives. By extracting both types of features simultaneously, the description of a scene will be simplified since edges corresponding to lines are not returned as edges anymore, but instead are included in the line width. Thus, a later reasoning step can be simplified since it will not have to sort out the edges that correspond to lines.

REFERENCES

Armande, N. and Montesinos, P., 1999. Thin nets extraction using a multi-scale approach. Computer Vision and Image Understanding 73(2), pp. 248–257.

Baumgartner, A., Steger, C., Mayer, H., Eckstein, W. and Ebner, H., 1999. Automatic road extraction based on multi-scale, grouping, and context. Photogrammetric Engineering & Remote Sensing 65(7), pp. 777–785.

Berzins, V., 1984. Accuracy of Laplacian edge detectors. Computer Vision, Graphics, and Image Processing 27, pp. 195–210.

Bignone, F., Henricsson, O., Fua, P. and Stricker, M., 1996. Automatic extraction of generic house roofs from high resolution aerial imagery. In: B. Buxton and R. Cipolla (eds), Fourth European Conference on Computer Vision, Lecture Notes in Computer Science, Vol. 1064, Springer-Verlag, Berlin, pp. 85–96.

Busch, A., 1996. A common framework for the extraction of lines and edges. In: International Archives of Photogrammetry and Remote Sensing, Vol. XXXI, part B3, pp. 88–93.

Canny, J., 1986. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(6), pp. 679–698.

Chen, J. S. and Medioni, G., 1989. Detection, localization, and estimation of edges. IEEE Transactions on Pattern Analysis and Machine Intelligence 11(2), pp. 191–198.

Cumani, A., 1991. Edge detection in multispectral images. Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing 53(1), pp. 40–51.

de Micheli, E., Caprile, B., Ottonello, P. and Torre, V., 1989. Localization and noise in edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 11(10), pp. 1106–1117.

Deriche, R., 1987. Using Canny's criteria to derive a recursively implemented optimal edge detector. International Journal of Computer Vision 1, pp. 167–187.

Di Zenzo, S., 1986. A note on the gradient of a multi-image. Computer Vision, Graphics, and Image Processing 33, pp. 116–125.

Dougherty, S. and Bowyer, K. W., 1998. Objective evaluation of edge detectors using a formally defined framework. In: K. J. Bowyer and P. J. Phillips (eds), Empirical Evaluation Methods in Computer Vision, IEEE Computer Society Press, Los Alamitos, CA, pp. 211–234.

Eberly, D., Gardner, R., Morse, B., Pizer, S. and Scharlach, C., 1994. Ridges for image analysis. Journal of Mathematical Imaging and Vision 4, pp. 353–373.

Fischer, A., Kolbe, T. H. and Lang, F., 1997. Integration of 2d and 3d reasoning for building reconstruction using a generic hierarchical model. In: W. Förstner and L. Plümer (eds), Semantic Modeling for the Acquisition of Topographic Information from Images and Maps, Birkhäuser Verlag, Basel, pp. 159–180.

Förstner, W., 1994. A framework for low level feature extraction. In: J.-O. Eklundh (ed.), Third European Conference on Computer Vision, Lecture Notes in Computer Science, Vol. 801, Springer-Verlag, Berlin, pp. 383–394.

Fritsch, D. and Sester, M., 1994. Test on image understanding. In: International Archives of Photogrammetry and Remote Sensing, Vol. XXX, part 3/1, pp. 243–248.

Fuchs, C., 1998. Extraktion polymorpher Bildstrukturen und ihre topologische und geometrische Gruppierung. Reihe C, No. 502, Deutsche Geodätische Kommission, München.

Haralick, R. M., 1984. Digital step edges from zero crossings of second directional derivatives. IEEE Transactions on Pattern Analysis and Machine Intelligence 6(1), pp. 58–68.

Heitger, F., 1995. Feature detection using suppression and enhancement. Technical Report TR–163, Communication Technology Laboratory, ETH Zürich, Switzerland.

Huertas, A. and Medioni, G., 1986. Detection of intensity changes with subpixel accuracy using Laplacian-Gaussian masks. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(5), pp. 651–664.

Iverson, L. A. and Zucker, S. W., 1995. Logical/linear operators for image curves. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(10), pp. 982–996.

Kisworo, M., Venkatesh, S. and West, G., 1994. Modeling edges at subpixel accuracy using the local energy approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(4), pp. 405–410.

Koenderink, J. J. and van Doorn, A. J., 1994. Two-plus-one-dimensional differential geometry. Pattern Recognition Letters 15(5), pp. 439–443.

Koller, T. M., Gerig, G., Székely, G. and Dettwiler, D., 1995. Multiscale detection of curvilinear structures in 2-d and 3-d image data. In: 5th International Conference on Computer Vision, pp. 864–869.

Lanser, S. and Eckstein, W., 1992. A modification of Deriche's approach to edge detection. In: 11th International Conference on Pattern Recognition, Vol. III, pp. 633–637.

Lanser, S., Zierl, C., Munkelt, O. and Radig, B., 1997. MORAL — a vision-based object recogntion system for autonomous mobile systems. In: Computer Analysis of Images and Patterns '97, Lecture Notes in Computer Science, Springer-Verlag, Berlin, pp. 33–41.

Lindeberg, T., 1998. Edge detection and ridge detection with automatic scale selection. International Journal of Computer Vision 30(2), pp. 117–156.

Maintz, J. B. A., van den Elsen, P. A. and Viergever, M. A., 1996. Evaluation of ridge seeking operators for multimodality medical image matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(4), pp. 353–365.

Marr, D. and Hildreth, E., 1980. Theory of edge detection. Proceedings of the Royal Society of London, Series B 207, pp. 187–217.

Nalwa, V. S. and Binford, T. O., 1986. On detecting edges. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(6), pp. 699–714.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P., 1992. Numerical Recipes in C: The Art of Scientific Computing. 2nd edn, Cambridge University Press, Cambridge.

Rothwell, C., Mundy, J., Hoffman, B. and Nguyen, V.-D., 1994. Driving vision by topology. Rapport de Recherche 2444, INRIA, Sophia Antipolis.

Shah, M., Sood, A. and Jain, R., 1986. Pulse and staircase edge models. Computer Vision, Graphics, and Image Processing 34, pp. 321–343.

Shen, J. and Castan, S., 1992. An optimal linear operator for step edge detection. Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing 54(2), pp. 112–133.

Shin, M. C., Goldgof, D. and Bowyer, K. W., 1998. An objective comparison methodology of edge detection algorithms using a structure from motion task. In: K. J. Bowyer and P. J. Phillips (eds), Empirical Evaluation Methods in Computer Vision, IEEE Computer Society Press, Los Alamitos, CA, pp. 235–254.

Sochen, N., Kimmel, R. and Malladi, R., 1998. A general framework for low level vision. IEEE Transactions on Image Processing 7(3), pp. 310–318.

Steger, C., 1997. Removing the bias from line detection. In: Computer Vision and Pattern Recognition, pp. 116–122.

Steger, C., 1998a. Analytical and empirical performance evaluation of subpixel line and edge detection. In: K. J. Bowyer and P. J. Phillips (eds), Empirical Evaluation Methods in Computer Vision, IEEE Computer Society Press, Los Alamitos, CA, pp. 188–210.

Steger, C., 1998b. An unbiased detector of curvilinear structures. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(2), pp. 113–125.

Steger, C., 1998c. Unbiased Extraction of Curvilinear Structures from 2D and 3D Images. Dissertation, Fakultät für Informatik, Technische Universität München. Herbert Utz Verlag, München.

Ulupinar, F. and Medioni, G., 1990. Refining ridges detected by a LoG operator. Computer Vision, Graphics, and Image Processing 51, pp. 275–298.

Wiedemann, C., Heipke, C., Mayer, H. and Hinz, S., 1998. Automatic extraction and evaluation of road networks from MOMS-02 imagery. In: International Archives of Photogrammetry and Remote Sensing, Vol. XXXII Part 3/1, pp. 285–291.