Extraction of Curved Lines from Images

Carsten Steger
Forschungsgruppe Bildverstehen (FG BV)
Informatik IX, Technische Universität München
Orleansstr. 34, 81667 München, Germany
E-mail: stegerc@informatik.tu-muenchen.de

Abstract

In this paper a method to extract curvilinear structures and their widths from digital images is presented. The approach is based on differential geometric properties of the image function. For each pixel, the second order Taylor polynomial is computed by convolving the image with the derivatives of a Gaussian smoothing kernel. Line points are required to have a vanishing gradient and a high curvature in the direction perpendicular to the line. The resulting filter generates a single response for each line. The line position can be determined with sub-pixel accuracy and the algorithm scales to lines of arbitrary width. A procedure to determine the width of the lines is described. It is based on locating the corresponding edge points of each line point in the direction perpendicular to the line with sub-pixel accuracy.

1. Introduction

Extracting lines in digital images is an important low-level operation in computer vision that has many applications, especially in photogrammetric and remote sensing tasks. There it can be used to extract linear features, like roads, railroads, or rivers, from satellite or low resolution aerial imagery. For some of these mapping tasks the extraction of the line width can be quite important.

The published schemes to line detection can be classified into three categories. The first approach detects lines by only considering the gray values of the image [3, 6]. Line points are extracted by using purely local criteria, e.g., local gray value differences. Since this will generate a lot of false hypotheses for line points, elaborate and computationally expensive perceptual grouping schemes have to be used to select salient lines in the image [2, 6]. These approaches usually assume lines to have a certain maximum width. Therefore, the line width is not extracted. Furthermore, lines cannot be extracted with sub-pixel accuracy.

The second approach is to regard lines as objects having parallel edges [7, 9]. In a first step, the local direction of a line is determined for each pixel. Then two tuned edge detection filters are used to extract the edges of the line and combined non-linearily [7]. The advantage of this approach is that since the edge detection filters are based on the derivatives of Gaussian kernels, the procedure can be iterated over the scale-space parameter σ to detect lines of arbitrary widths. By maximizing the response in scale-space, a rough estimate of the line width can be obtained. However, because special directional edge detection filters have to be constructed that are not separable, the approach is computationally expensive.

In the third approach, the image is regarded as a function $z(x, y)$ and lines are detected as ridges and ravines in this function by locally approximating the image by its second or third order Taylor polynomial. The coefficients of this polynomial are usually determined by using the facet model, i.e., by a least squares fit of the polynomial to the image data over a window of a certain size [4, 1, 5]. The direction of the line is determined from the Hessian matrix of the Taylor polynomial. Line points are then found by selecting pixels that have a high second directional derivative, i.e., a high curvature, perpendicular to the line direction. The advantage of this approach is that lines can be detected with sub-pixel accuracy without constructing specialized directional filters. However, this approach usually leads to multiple responses to a single line, especially when masks larger than 5×5 are used to suppress noise [8, 1]. Furthermore, these approaches do not attempt to extract the line width.

In this paper an approach to line detection that uses the differential geometric approach of the third category of operators will be presented. In contrast to those, the coefficients of a second order Taylor polynomial are determined by convolving the image with the derivatives of a Gaussian smoothing kernel. Because of this, the algorithm can be scaled to lines of arbitrary width. Additionally, an algorithm to determine the width of the line for each line point is presented.

2. Detection of Line Points

2.1. Models for Lines in 1D

Many approaches to line detection consider lines in 1D to be bar-shaped, i.e., the ideal line of width $2w$ and height h is assumed to have a profile given by

$$f_b(x) = \begin{cases} h, & |x| \leq w \\ 0, & |x| > w \end{cases} . \quad (1)$$

However, due to sampling effects of the sensor lines usually do not have this profile (see [8]). Therefore, in this paper lines are assumed to have an approximately parabolic profile. The ideal line of width $2w$ and height h is then given by

$$f_p(x) = \begin{cases} h\big(1-(x/w)^2\big), & |x| \leq w \\ 0, & |x| > w \end{cases} . \quad (2)$$

The line detection algorithm will be developed for this type of profile, but the implications of applying it to bar-shaped lines will be considered later on.

2.2. Detection of Lines in 1D

In order to detect lines with a profile given by (2) in an image $z(x)$ without noise, it is sufficient to determine the points where $z'(x)$ vanishes. However, it is usually convenient to select only salient lines. A useful criterion for salient lines is the magnitude of the second derivative $z''(x)$ in the point where $z'(x) = 0$. Bright lines on a dark background will have $z''(x) \ll 0$ while dark lines on a bright background will have $z''(x) \gg 0$.

Real images will contain a significant amount of noise. Therefore, the scheme described above is not sufficient. In this case, the first and second derivatives of $z(x)$ should be estimated by convolving the image with the derivatives of the Gaussian smoothing kernel

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} . \quad (3)$$

The responses, i.e., the estimated derivatives, are:

$$r_p(x,\sigma,w,h) = g_\sigma(x) * f_p(x) \quad (4)$$
$$r'_p(x,\sigma,w,h) = g'_\sigma(x) * f_p(x) \quad (5)$$
$$r''_p(x,\sigma,w,h) = g''_\sigma(x) * f_p(x) . \quad (6)$$

The detailed equations can be found in [8].

Equations (4)–(6) give a complete scale-space description of the ideal line profile f_p when it is convolved with the derivatives of Gaussian kernels. It is apparent from these equations that $r'_p(x,\sigma,w,h) = 0 \Leftrightarrow x = 0$ for all σ. Furthermore, $r''_p(x,\sigma,w,h)$ takes on its maximum negative value at $x = 0$ for all σ. Hence it is possible to determine the precise location of the line for all σ.

For a bar profile without noise no simple criterion that depends only on $z'(x)$ and $z''(x)$ can be given since $z'(x)$ and $z''(x)$ vanish in the interval $[-w, w]$. However, if the bar profile is convolved with the derivatives of the Gaussian kernel, a smooth function is obtained. The responses are:

$$r_b(x,\sigma,w,h) = h\big(\phi_\sigma(x+w) - \phi_\sigma(x-w)\big) \quad (7)$$
$$r'_b(x,\sigma,w,h) = h\big(g_\sigma(x+w) - g_\sigma(x-w)\big) \quad (8)$$
$$r''_b(x,\sigma,w,h) = h\big(g'_\sigma(x+w) - g'_\sigma(x-w)\big) . \quad (9)$$

It can be seen that bar profile gradually becomes "round" at its corners the larger σ is chosen. The first derivative will vanish only at $x = 0$ for all $\sigma > 0$ because of the infinite support of $g_\sigma(x)$. However, the second derivative $r''_b(x,\sigma,w,h)$ will not take on its maximum negative value for small σ. Furthermore, there will be two distinct minima in the interval $[-w, w]$. It is, however, desirable for $r''_b(x,\sigma,w,h)$ to exhibit a clearly defined minimum at $x = 0$. It can be shown that

$$\sigma \geq w/\sqrt{3} \quad (10)$$

has to hold for this. Furthermore, it can be shown that $r''_b(x,\sigma,w,h)$ will have its maximum negative response in scale-space for $\sigma = w/\sqrt{3}$. This means that the same scheme as described above can be used to detect bar-shaped lines as well. However, the restriction on σ must be observed. The same analysis could be carried out for other types of lines as well, e.g., roof-shaped lines. However, it is expected that no fundamentally different results will be obtained. For all σ above a certain value that depends on the line type the responses will show the desired behaviour of $z'(0) = 0$ and $z''(0) \ll 0$ with $z''(x)$ having a distinct minimum.

2.3. Lines in 1D, Discrete Case

The analysis so far has been carried out for analytical functions $z(x)$. For discrete signals only two modifications have to be made. The first one is the choice of how to implement the convolution in discrete space. Integrated Gaussian kernels were chosen as convolutions masks, mainly because they give automatic normalization of the masks and a direct criterion on how many coefficients are needed for a given approximation error. The integrated Gaussian is obtained if one regards the discrete image z_n as a piecewise constant function $z(x) = z_n$ for $x \in (n-\frac{1}{2}, n+\frac{1}{2}]$ and integrating the continuous Gaussian kernel over this area. The convolution masks will be given by:

$$g_{n,\sigma} = \phi_\sigma(n+\tfrac{1}{2}) - \phi_\sigma(n-\tfrac{1}{2}) \quad (11)$$
$$g'_{n,\sigma} = g_\sigma(n+\tfrac{1}{2}) - g_\sigma(n-\tfrac{1}{2}) \quad (12)$$
$$g''_{n,\sigma} = g'_\sigma(n+\tfrac{1}{2}) - g'_\sigma(n-\tfrac{1}{2}) . \quad (13)$$

The approximation error is set to 10^{-4} in each case.

The second problem that has to be solved is how to determine the location of a line in the discrete case. In principle, one could use a zero crossing detector for this task. However, this would yield the position of the line only with pixel accuracy. In order to overcome this, the second order Taylor polynomial of z_n is examined. Let r, r', and r'' be the locally estimated derivatives at point n of the image that are obtained by convolving the image with g_n, g'_n, and g''_n. Then the Taylor polynomial is given by

$$p(x) = r + r'x + \frac{1}{2}r''x^2 \ . \tag{14}$$

The position of the line, i.e., the point where $p'(x) = 0$ is

$$x = -\frac{r'}{r''} \ . \tag{15}$$

The point n is declared a line point if this position falls within the pixel's boundaries, i.e., if $x \in [-\frac{1}{2}, \frac{1}{2}]$ and the second derivative r'' is larger than a user-specified threshold. Please note that in order to extract lines, the response r, which is the smoothed local image intensity, is unnecessary and therefore does not need to be computed.

2.4. Detection of Lines in 2D

Curvilinear structures in 2D can be modeled as curves $s(t)$ that exhibit a characteristic 1D line profile (e.g., f_p or f_b) in the direction perpendicular to the line, i.e., perpendicular to $s'(t)$. Let this direction be $n(t)$. This means that the first directional derivative in the direction $n(t)$ should vanish and the second directional derivative should be of large absolute value. No assumption can be made about the derivatives in the direction of $s'(t)$. For example, let $z(x, y)$ be an image that results from sweeping the profile f_p along a circle $s(t)$ of radius r. The second directional derivative perpendicular to $s'(t)$ will have a large negative value, as desired. However, the second directional derivative along $s'(t)$ will also be non-zero.

The only problem that remains is to compute the direction of the line locally for each image point. In order to do this, the partial derivatives r_x, r_y, r_{xx}, r_{xy}, and r_{yy} of the image will have to be estimated. This can be done by convolving the image with the appropriate 2D Gaussian kernels. The direction in which the second directional derivative of $z(x, y)$ takes on its maximum absolute value will be used as the direction $n(t)$. This direction can be determined by calculating the eigenvalues and eigenvectors of the Hessian matrix

$$H(x, y) = \begin{pmatrix} r_{xx} & r_{xy} \\ r_{xy} & r_{yy} \end{pmatrix} \ . \tag{16}$$

The calculation can be done in a numerically stable and efficient way by using one Jacobi rotation to annihilate the r_{xy} term. Let the eigenvector corresponding to the eigenvalue of maximum absolute value, i.e., the direction perpendicular to the line, be given by (n_x, n_y) with $\|(n_x, n_y)\|_2 = 1$. As in the 1D case, a quadratic polynomial will be used to determine whether the first directional derivative along (n_x, n_y) vanishes within the current pixel. This point will be given by

$$(p_x, p_y) = (tn_x, tn_y) \ , \tag{17}$$

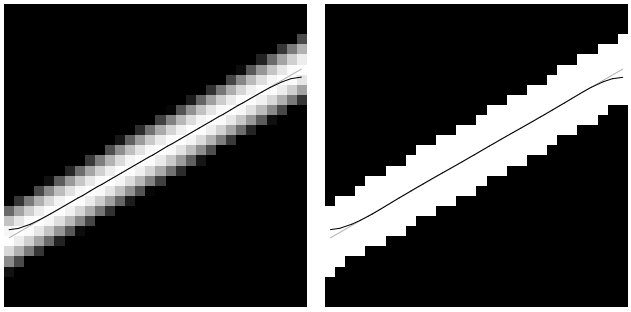
where

$$t = -\frac{r_x n_x + r_y n_y}{r_{xx} n_x^2 + 2r_{xy} n_x n_y + r_{yy} n_y^2} \ . \tag{18}$$

Again, $(p_x, p_y) \in [-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$ is required in order for a point to be declared a line point. As in the 1D case, the second directional derivative along (n_x, n_y), i.e., the maximum eigenvalue, can be used to select salient lines.

2.5. Examples

Figure 1 gives an example of the results obtainable with the presented approach. Here, bright lines were extracted from an image of an ideal parabolic line (Fig. 1(a)) and a bar-shaped line (Fig. 1(b)) and were linked into contours using the algorithm given in [8]. In both cases the line width is 6 pixels ($w = 3$) and the angle of the line is 30°. The true position of the line is indicated by the medium gray line. It is evident that the detected line is in exactly the right position, except at the borders of the image, where in this implementation the pixel values were mirrored.



(a) Ideal Parabolic Line (b) Bar-Shaped Line

Figure 1. Lines detected in images of an ideal parabolic line (a) and a bar-shaped line (b) ($\sigma = 1.8$)

3. Determination of the Line Width

In many applications it is entirely sufficient to know the exact position of the line in an image. However, often it is

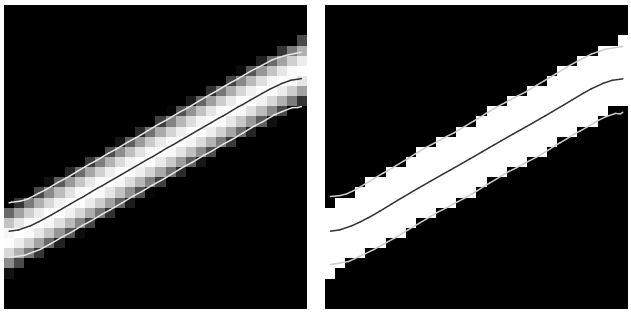
also very important to know the width of the line for each line point. Therefore, this section will present an approach to solve this problem.

According to (1) and (2), both the parabolic line and the bar-shaped line will exhibit a maximum in the absolute value of the gradient at the edges of the line. Hence, to detect the width of the line, for each line point the closest points in the image (to the left and to the right of the line point) where the absolute value of the gradient takes on its maximum value need to be determined. Of course, these points should be searched for exclusively along a line in the direction perpendicular to the current line. Only a trivial modification of the Bresenham line drawing algorithm is necessary to yield all pixels that this line will intersect. The analysis in Sect. 2.2 shows that it is sensible to search for edges only in a restricted neighbourhood of the line. Ideally, the line to search would have a length of $\sqrt{3}\sigma$. In order to ensure that most of the edge points are detected, the current implementation uses a slightly larger line length of 2.5σ.

In an image of the absolute value of the gradient of the image the desired edges will appear as bright lines. Hence, the algorithm to detect line points described Sect. 2 could in principle be used on the gradient image to detect the edges of the line with sub-pixel accuracy. However, this would mean that some additional smoothing would be applied to the gradient image. This is undesireable since it would destroy the correlation between the location of the line points and the location of the corresponding edge points. Therefore, the edge points in the gradient image are extracted with a facet model line detector which uses the same principles as described in Sect. 2 but uses different convolution masks to determine the partial derivatives of the image [1, 4, 8]. The smallest possible mask size (3×3) is used since this will result in the most accurate localization of the edge points while yielding as little of the problems mentioned in Sect. 1 as possible. It has the additional benefit that the computational costs are relatively low.

Figure 2 shows the result of applying this approach to the images of Fig. 1. The detected lines are shown as dark contours in the image, while the corresponding edge points are displayed as light contours. The line in Fig. 2(a) is an parabolic line with $w = 3$. The algorithm estimates the line width to be ≈ 2.46. For the bar-shaped line with $w = 3$ shown in Fig. 2(b) the estimated line width is ≈ 3.08. In both cases this is a very good approximation of the real line width, especially when the discrete nature of the image is taken into account.

Considering (5) and (8) again, it is not surprising that for the parabolic line the width is estimated too low. The estimated width of the line strongly depends on the values of σ and w. For values of σ lying in a range between 0 and $\approx 0.85w$ the width will be estimated slightly too low for this type of profile, but never less than $\approx$ 70% of the real line



(a) Ideal Parabolic Line (b) Bar-Shaped Line

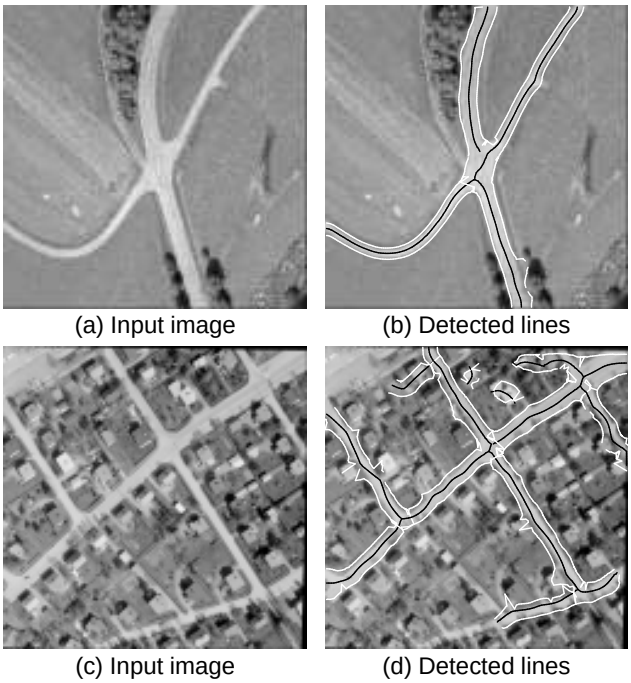
Figure 2. Lines and line widths in images of an ideal parabolic line (a) and a bar-shaped line (b) ($\sigma = 1.8$)

width. For higher values of σ the estimated line width will be higher than the real line width. For bar-shaped lines the width of the line will never be estimated too low. Its width will be very close to the actual line width in a range between 0 and $\approx w/\sqrt{3}$. Again, for large values of σ the width will be estimated too high. It will be interesting to see whether the estimated line widths can be adjusted to the real line widths by using (5) and (8). These equations give a prediction on where the corresponding edges of a line will be estimated as a function depending on σ and w. In principle, the inverse of this function could be used to correct the estimated line widths to the true line widths. Further research will focus on this topic.

4. Further Examples

In this section two more examples of the versatility of the proposed approach will be given. Figures 3(a) and 3(c) show aerial images of different resolutions. The algorithm was applied with thresholds based on the expected maximum line width and minimum contrast, according to the scale-space analysis given in [8]. It can be seen from Fig. 3(b) and 3(d) that the algorithm is able to extract most of the salient lines from the images. As in Sect. 3, dark contours are used to display the line position and light contours to display the line width.

The locations of the detected lines in Fig. 3(b) are very accurate, and the detected line width corresponds closely to the real line width. The only problematic area is the lower middle part of the image where nearby strong edges result in the line width to be estimated to large. For the image in Fig. 3(d) the locations of the lines are fairly accurate as well, even in those parts of the image where trees are casting shadows onto the roads. One exception is the bottommost



(a) Input image (b) Detected lines

(c) Input image (d) Detected lines

Figure 3. Lines detected in aerial images

line in the image which is detected in a slightly shifted location because it has strongly differing gray values on each side of the line. This effect is to be expected and has been analyzed in detail in [8]. Furthermore, it can be seen that the algorithm is able to extract lines of varying width without problems. The estimated widths are very close to the real widths of the lines. In some cases, however, an edge point corresponding to a line point has not been found due to the fact that the line to search for edges had limited length. Furthermore, some very strong edges again cause the line width to be estimated too high in some places. The reason for this is that in real images the strongest edge perpendicular to the line direction does not always correspond to the true line edge and therefore the witdh is wrongly estimated. If the line width is assumed to have some regularity an outlier detection mechanism might be useful to detect and correct such cases.

5. Conclusions

In this paper a low-level approach to the extraction of curvilinear structures and their widths from images was presented. The advantages of this approach are that line extraction is done using only the first and second directional derivatives of the image. In contrast to [7], no specialized directional filters are needed. This makes the approach computationally efficient. Furthermore, since the derivatives are estimated by convolving the image with the derivatives of a Gaussian smoothing kernel, only a single response is generated for each line.

An algorithm has been presented to extract the width of the line for each line point. The algorithm is computationally efficient since it looks for the edge points of a line starting at the line points already found and using a search space of fixed size. Again, no specialized directional filters are needed. The algorithm yields the sub-pixel position of the left and right edges. For bar-shaped lines the estimate of the width is very precise, while for parabolic shaped lines the widths are slightly underestimated in some cases. Further research will focus on whether it is possible to correct the estimate in such cases.

References

[1] A. Busch. Fast recognition of lines in digital images without user-supplied parameters. In H. Ebner, C. Heipke, and K. Eder, editors, *Spatial Information from Digital Photogrammetry and Computer Vision*, volume 30, part 3/1 of *International Archives of Photogrammetry and Remote Sensing*, pages 91–97, 1994.

[2] M. A. Fischler. The perception of linear structure: A generic linker. In *Image Understanding Workshop*, pages 1565–1579, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers.

[3] M. A. Fischler, J. M. Tenenbaum, and H. C. Wolf. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. *Computer Graphics and Image Processing*, 15:201–223, 1981.

[4] F. Glazer. Curve finding by ridge detection and grouping. In W. Kropatsch and H. Bischof, editors, *Mustererkennung*, Informatik Xpress 5, pages 109–116. Deutsche Arbeitsgemeinschaft für Mustererkennung, 1994.

[5] R. M. Haralick, L. T. Watson, and T. J. Laffey. The topographic primal sketch. *International Journal of Robotics Research*, 2(1):50–72, 1983.

[6] B. Jedynak and J.-P. Rozé. Tracking roads in satellite images by playing twenty questions. In A. Gruen, O. Kuebler, and P. Agouris, editors, *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, pages 243–253, Basel, Switzerland, 1995. Birkhäuser Verlag.

[7] T. M. Koller, G. Gerig, G. Székely, and D. Dettwiler. Multiscale detection of curvilinear structures in 2-d and 3-d image data. In *Fifth International Conference on Computer Vision*, pages 864–869. IEEE Computer Society Press, 1995.

[8] C. Steger. Extracting curvilinear structures: A differential geometric approach. In B. Buxton and R. Cipolla, editors, *Fourth European Conference on Computer Vision*, volume 1064 of *Lecture Notes in Computer Science*, pages 630–641. Springer-Verlag, 1996.

[9] J. B. Subirana-Vilanova and K. K. Sung. Multi-scale vector-ridge-detection for perceptual organization without edges. A.I. Memo 1318, MIT Artificial Intelligence Laboratory, Cambridge, MA, USA, Dec. 1992.