

 Open access • Journal Article • DOI:10.1142/S0218001401001283

## Curvature in digital 2d images — Source link

Vladimir Kovalevsky

**Institutions:** University of Rostock

**Published on:** 01 Nov 2001 - International Journal of Pattern Recognition and Artificial Intelligence (World Scientific Publishing Company)

**Topics:** Curvature, Digital image processing and Subpixel rendering

Related papers:

- [Digital curvature estimation](#)
- [Multigrid Analysis of Curvature Estimators](#)
- [Image Blur Analysis for the Subpixel-level Measurement of In-plane Vibration Parameters of MEMS Resonators](#)
- [Edge Detection Performance in Super-Resolution Image Reconstruction from Camera Arrays](#)
- [Digital image correlation using Newton-Raphson method of partial differential correction](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/curvature-in-digital-2d-images-dvlqh44ynx>

# Curvature in Digital 2D Images

*Vladimir Kovalevsky*

University of Rostock

Institute of Computer Graphics

Albert-Einstein-Str. 21, 18051 Rostock, Germany

kovalev@tfh-berlin.de

## Abstract

The paper presents an analysis of sources of errors when estimating derivatives of numerical or noisy functions. A method of minimizing the errors is suggested. When being applied to the estimation of the curvature of digital curves, the analysis shows that under the conditions typical for digital image processing the curvature can rarely be estimated with a precision higher than 50%. Ways of overcoming the difficulties are discussed and a new method for estimating the curvature is suggested and investigated as to its precision. The method is based on specifying boundaries of regions in gray value images with sub-pixel precision. The method has an essentially higher precision than the known methods.

## 1 Introduction

The problem of calculating the curvature of a digital curve must be considered as estimating the curvature of its preimage which is supposed to be a smooth curve with a restricted curvature. Otherwise the solution of the problem is undefined.

Since the curvature is proportional to the second derivative  $d^2y/dx^2$ , the problem of estimating the curvature has much in common with that of estimating the derivatives of numerical functions, i.e. functions defined by a numerical algorithm. We shall show in Section 2 that an estimation of a derivative of a function whose values are specified with a limited precision (which is always the case in numerical computations) cannot be based on the classical definition of a derivative as a limit. We also shall present a method of minimizing the error of the estimate. The method may be directly used to minimize the error of estimating the

curvature of a digital curve whose preimage is known (or is supposed) to be a smooth curve. This approach is presented in Section 3.

Since the desired estimate is an estimate of the preimage curvature, any solution of the problem must be based on estimations of the locations of some *points of the preimage curve* starting with the data contained in the digital image. All known methods of estimating the curvature in digital images have a common drawback: in the most practically relevant cases the precision of the estimates of the locations of preimage points is not sufficient to get satisfactorily precise estimates of the curvature. The reason is the loss of information during the digitization of the image. Even in a high resolution image of some thousand scan lines there may occur an arc with a curvature radius of say 20 pixels whose length is of the same order of magnitude. The precision of estimating the coordinates of points of the preimage is in the best case of the order of half a pixel. According to the results of Section 3, it is impossible to estimate the curvature of such an arc better than with an error of  $\pm 40\%$ , which is mostly not sufficient.

One of the best published methods of estimating the curvature is based on averaging the positions of the points [WS93, Wor93]. As it is known from the mathematical statistics, averaging  $N$  random values with constant weights reduces the RMS error by the factor of  $\sqrt{N}$ . When averaging with decreasing weights, e.g. according to a Gaussian kernel, the factor of reducing the RMS error is still less. Decreasing the coordinate errors by the factor of  $\sqrt{N}$  is in many cases not sufficient. Really, for radii under 30 pixels and arcs of  $45^\circ$  the authors have reported an RMS error greater than 22% (Fig. 12 in [WS93]) which means a maximum error of about 66% plus a bias.

Such values of curvature errors are typical. Many attempts have been made to improve the results. Thus it was shown in [Via96] that the method of the Euclidean path, as suggested there, has a better precision. However, for a radius of 50 pixels and for arcs essentially shorter than  $45^\circ$ , as that used in [WS93], the RMS error is still up to 37%. This corresponds to a maximum error of about 100% which may be still greater due to the bias of up to 28% (Table 1 of [Via96]). For the radius of 10 pixels the author reports an essentially smaller RMS error, namely up to 8.6%, which may be explained as follows. The same number of points (ca. 19 points in the case of the last row and last column of Tables 1 and 2 of [Via96]) compose an arc of only about  $22^\circ$  in the case of radius 50 and of  $108^\circ$  in the case of radius 10. The arc with a greater angle enables a higher precision of curvature estimation.

In [Hla94] a bias up to 492% due to systematic errors, when using the approach by [Low89], is reported. The authors have improved this method and report an error in estimating the curvature, which does not exceed 30%. However, it is the *error of the mean* of measurements performed at all points of a synthetic digital circle. There are no data about the errors of single measurements reported.

Thus we may conclude, that there are no known methods of estimating the curvature in digital images with an error not greater than say 20% and applicable in the very common case when the preimage of the curve is an arc of less than  $45^\circ$  with a radius less than 20 pixels.

To understand the cause of the difficulties we analyze at first, in Section 2, the possibility to increase the precision of estimating derivatives, including the second derivative, which is of importance for estimating the curvature. In Section 3 we consider the precision of estimating the curvature. We came to the conclusion, that the only way to increase the precision consists in increasing the precision of estimating the coordinates of preimage points. Therefore we consider in Section 4 the possibility of using the information about the location of the boundary of an object which is represented by a *gray value image*.

An attempt to use this information is presented in [DF90]. The authors find the boundary of an object by means of Canny's edge detector. Then they estimate the gradient of the gray values near the boundary and use the finite difference of the gradients at two points lying close to each other to estimate the curvature of the boundary. We are using here another approach.

Consider an analogous (i.e. not digitized) image  $AI$  containing a region with a high constant brightness against a dark background. During the digitization of  $AI$  some pixels of the raster are only partially covered by the light region. Such a pixel gets an intermediate gray level depending upon the area of its intersection with the region. Thus a gray value image  $GVI$  occurs. We solve the problem of reconstructing  $AI$  given the image  $GVI$ . Of course, it is only possible under certain suppositions about  $AI$ , e.g. that the boundaries of its regions are smooth curves with a restricted curvature. Thus we are looking for an image  $AI$  whose object boundaries have the minimal curvature while the result of digitizing  $AI$  is equal to  $GVI$ . The curvature of the boundaries of  $AI$  is then considered as the curvature estimates at the corresponding points of  $GVI$ . These values may be then transferred, if necessary, to the binary image resulting from the binarization of  $GVI$ . The solution of this problem is presented in Section 4; experimental results are given in Section 5. Section 6 contains the conclusions and a discussion of the perspectives.

## 2 Optimal estimation of the derivatives of numerical functions.

### 2.1 The first derivative

The first derivative of a function  $f(x)$  is, according to its definition, the limit of (1) as  $dx$  tends to zero. Therefore one may tend to believe that to compute the derivative of a numerical function, e.g. of one given as a computer subroutine, with the maximum precision, one should compute the value

$$(f(x+dx)-f(x))/dx \quad (1)$$

with as small value of  $dx$  as possible. However, this belief is erroneous. We shall demonstrate at first a simple example and then analyze the essence of the problem. Fig. 1 shows the results of computing the value of (1) for the case  $f(x)=x^2$  in the interval  $[1.3, 1.7]$  with the values of  $dx=10^{-p}$ , with  $p=16, 17, 18$  and  $19$ . As one may see, the results become at first inaccurate and then absolutely crazy.

The reason is the limited precision of representing the value of  $f(x)$  in the computer, which may be very high, however, still finite.

To investigate this phenomenon consider the influence of errors in representing  $f(x)$  onto the value of (1):

$$\text{Estimate}(f'(x)) = (f(x+dx) + er_1 - f(x) - er_2) / dx = (f(x+dx) - f(x)) / dx + (er_1 - er_2) / dx; \quad (2)$$

where  $er_1$  and  $er_2$  are the errors in computing  $f(x+dx)$  and  $f(x)$  respectively.

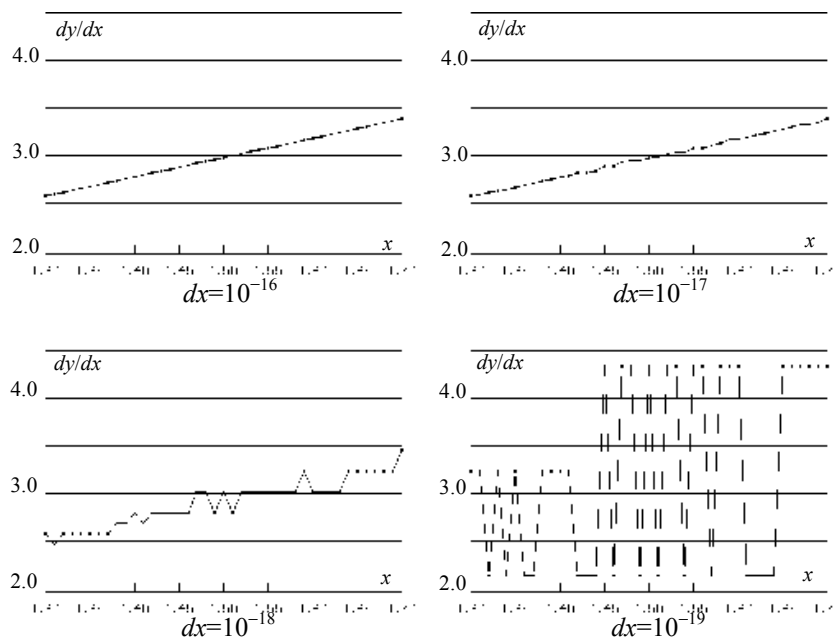


Fig. 1. Estimating the first derivative of  $y=x^2$  as  $dy/dx$  with very small values of  $dx$

If these errors are limited by the value of  $\varepsilon$  then the worst case error  $E$  of the estimate is equal to

$$Er = 2 \cdot \varepsilon / dx. \quad (3)$$

As soon as  $dx$  becomes of the order of magnitude of  $\varepsilon$  the error  $Er$  of the estimate becomes unacceptably great.

According to (3) the error  $Er$  may be made arbitrarily small by increasing  $dx$ . However, there is another source of errors, which must be taken into account: if the desired derivative is not constant (i.e.  $f(x)$  is not a linear function) then the estimate (1) brings the *average* derivative in the interval  $[x, x+dx]$ , which may be essentially different from the value of the derivative at the point  $x$ . It is known that the estimate (1) is equal to the exact value of the derivative at some unknown point in the said interval. When increasing  $dx$  we increase the precision of the desired estimate, however, we simultaneously decrease the precision of the *location* of the estimate. Thus equation (3) represents some kind of *principle of uncertainty* (which may have much in common with the famous Heisenberg uncertainty principle in the quantum mechanics): the greater the precision of the location of the estimate the smaller the precision of the estimate itself.

It is easy to see that there is an *optimum value* of  $dx$  yielding the highest precision. We consider here the *maximum possible error* of the estimate rather than an RMS error because knowing the maximum error gives us a *guaranty* that the actual error will never override the found limit.

To find the optimum value of  $dx$  consider the worst case error of the estimate (2) while representing  $f(x)$  by its Taylor series up to the second order term:

$$\text{Estimate}(f'(x))=(f(x+dx)+er_1-f(x)-er_2)/dx=f(x)+0.5\cdot f''(x+k\cdot dx)\cdot dx+(er_1-er_2)/dx;$$

where  $k$  is a value between 0 and 1.

The error  $Er$  of the estimate is then equal to:

$$Er=\text{Estimate}(f'(x))-f'(x)=0.5\cdot f''(x+k\cdot dx)\cdot dx+(er_1-er_2)/dx;$$

To get the maximum absolute error we should substitute the maximum value  $F2$  of  $f''(x)$  in the interval  $[x, x+dx]$  for  $f''(x+k\cdot dx)$ ,  $\varepsilon$  for  $er_1$  and  $-\varepsilon$  for  $er_2$ :

$$\max Er=0.5\cdot F2\cdot dx+2\cdot \varepsilon /dx; \quad (4)$$

To get the optimal value of  $dx$  we must find the minimum of (4) with respect to  $dx$ , i.e. to set the partial derivative of (4) with respect to  $dx$  equal to 0 and to solve the corresponding equation. It follows:

$$\text{optimal } dx=2\cdot \sqrt{\varepsilon /F2}; \quad (5)$$

where  $\varepsilon$  is the maximum possible error of specifying the values of  $f(x)$  and  $F2$  is the maximum possible value of the second derivative of  $f(x)$  in the interval  $[x, x+dx]$ . Since we are mostly interested to know and to minimize the *order of magnitude* of the maximum error (4) rather than its exact value, the parameters  $\varepsilon$  and  $F2$  in the right hand side of (5) may be estimated rather coarsely.

To illustrate the usage of (5) let us apply it to the case of Fig. 1. When computing  $f(x)$  as a double precision float with a mantissa of 53 bits the value  $\varepsilon$  is equal to the rounding error and is of the order of  $2^2\cdot 2^{-54}=2^{-52}\cong 10^{-16}$ . Note that this is not the smallest number representable in the computer, which is much less. The smallest number is specified by the exponent, rather than by the mantissa. The optimal value of  $dx$  is, according to (5) of the order of  $10^{-8}$ . The error in computing the derivative at  $x=1.5$  becomes unacceptably great when  $dx$  becomes comparable with or smaller than  $2\cdot \varepsilon =2\cdot 10^{-16}$  (since the second term in (4) becomes greater than 1.0), which situation we actually see in Fig. 1.

The error may be made still smaller if we use a symmetrical estimate of the first derivative:

$$\text{Estimate1S}=(f(x+dx)-f(x-dx))/(2\cdot dx); \quad (6)$$

The optimal value of  $dx$  may be deduced in the same way as before, however, the Taylor series should be prolonged until the term with the third derivative, since the terms with the second derivative disappear.

Then we obtain:

$$\text{Error}=(1/6)\cdot F3\cdot dx^2+\varepsilon/dx; \quad (7)$$

and

$$\text{optim } dx=(3\cdot \varepsilon/F3)^{1/3}; \quad (8)$$

where  $F3$  is the estimate of the maximum absolute value of the third derivative of  $f(x)$  in the interval  $[x-dx, x+dx]$ . A similar deduction may be found in [Bu92].

It is important to notice that in the case  $F3=0$  the value of  $dx$  should be chosen as great as possible. To make this clear, the reader may persuade himself, that the estimate (6) yields the *precise value* of the first derivative at the point  $x$  with any value of  $dx$  if  $f(x)$  is a polynom of degree 2.

Now we can specify the worst case error when using the optimal value of  $dx$ . We put the value (8) into (7) and obtain:

$$\text{minimax Error} = ((1/6) \cdot 3^{2/3} + 3^{-1/3}) \cdot \epsilon^{2/3} \cdot F3^{1/3} \cong 1.04 \cdot \epsilon^{2/3} \cdot F3^{1/3}. \quad (9)$$

## 2.2 The second derivative

The estimate of the second derivative of a function whose values are specified with a limited precision may be derived in a similar way. We consider as the estimate the well-known expression for the second difference:

$$\text{Estimate2} = (f(x+dx) - 2 \cdot f(x) + f(x-dx)) / dx^2; \quad (10)$$

The worst case error of the numerator of (10) is obviously equal to  $4 \cdot \epsilon$ , where  $\epsilon$  is as before the maximum error of calculating the values of  $f(x)$ . After having replaced in (10)  $f(x+dx)$  and  $f(x-dx)$  by Taylor series up to the terms with the fourth derivative one obtains the value of the maximum error as:

$$\text{Error} = (1/12) \cdot F4 \cdot dx^2 + 4 \cdot \epsilon / dx^2; \quad (11)$$

where  $F4$  is the estimate of the upper bound of the absolute value of the fourth derivative of  $f(x)$  in the interval  $[x-dx, x+dx]$ . By setting the partial derivative of (11) with respect to  $dx^2$  equal to 0 we obtain the optimal value of  $dx$ :

$$\text{optim } dx = (48 \cdot \epsilon / F4)^{1/4} \quad (12)$$

and the minimax error:

$$\text{minimax Error} = ((1/12) \cdot 48^{1/2} + 4 \cdot 48^{-1/2}) \cdot \epsilon^{1/2} \cdot F4^{1/2} \cong 1.15 \cdot (\epsilon \cdot F4)^{1/2}. \quad (13)$$

The values of  $F3$  (respectively  $F4$ ) may be estimated rather coarsely since the value of (8) (respectively (12)) is a minimum and it depends upon  $F3$  (respectively  $F4$ ) as a rather low power. Thus, it may be shown that when multiplying  $F4$  by a factor of 3.0 the value of (12) changes only by 50% and that of (13) by 34%.

Coarse estimates of the values of the third and the fourth derivatives, which one needs for the calculation of the above expressions, may be also obtained by means of symmetric finite differences of the corresponding order:

$$F3 \cong (f(x+2 \cdot dx) - 2 \cdot f(x+dx) + 2 \cdot f(x-dx) - f(x-2 \cdot dx)) / (2 \cdot dx^3); \quad (14)$$

$$F4 \cong (f(x+2 \cdot dx) - 4 \cdot f(x+dx) + 6 \cdot f(x) - 4 \cdot f(x-dx) + f(x-2 \cdot dx)) / dx^4; \quad (15)$$

where  $dx$  is the half length of the interval in which these values are supposed to retain their order of magnitude.

### 3 Estimates of the curvature

The well-known expression for the curvature of a curve specified by  $y=f(x)$  is:

$$C=f''(x)/(1+f'(x)^2)^{3/2}; \quad (16)$$

Unfortunately, it is not so easy to find the optimal estimate for this expression in the way described above. Instead we consider a coordinate system, whose  $X$ -axis is the tangent to the curve at the point with a given  $x$ , and the  $Y$ -axis is the normal. Then the first derivative  $f'(x)$  becomes zero in this coordinate system and the curvature becomes equal to the second derivative:

$$C=f''(x); \quad (17)$$

Thus we can use the estimates (12) and (13) to approximately specify the optimal value of  $dx$  and to get an idea of the precision which may be reached. To estimate the curvature of a given digital curve (i.e. the minimum curvature of its preimage) we must take three points on the curve, whose projections onto the tangent are spaced by the value  $dx$  calculated by (12). Then we may expect to estimate the curvature with an error not exceeding the value (13).

By means of these estimates we can demonstrate that it is rather difficult (or perhaps impossible) to reach practically acceptable estimates of the curvature in binary images. In such images the maximum error in measuring the coordinates is on the order of 0.7 pixel. Suppose, we want the guarantee that the minimax error  $dC$  not exceeds a modest value of 10%. If the curve may be supposed to have the fourth derivative on the same order of magnitude as a circle of the same curvature, i.e.  $F4 \approx 3 \cdot C^3$ , then according to (17) and (13) one arrives at the following demand:

$$dC/C = 1.15 \cdot (0.7 \cdot 3 \cdot C^3)^{1/2} / C \leq 0.1 \text{ implies } C \leq 0.0036 \text{ pixel}^{-1};$$

which corresponds to a radius of at least 277 pixels. When calculating the optimal  $dx$ , for which this precision may be reached, one arrives by (12) at  $dx = (48 \cdot 0.7/3)^{1/4} \cdot C^{-3/4} = 125$  pixels which corresponds to an arc of  $53^\circ$ . In the cases of curves with greater curvatures the error is always greater than 10%. For instance, for  $C = 1/16 \approx 0.063$  (i.e.  $R = 16$ ) and  $F4 = 3 \cdot C^3$  the minimax error is according to (13) equal to  $1.15 \cdot \sqrt{0.7 \cdot 3 \cdot 16^{-3}} \approx 0.024$  which corresponds to 38% of the value of  $C$ . This rather modest precision may be reached if the value of  $dx$  may be chosen near the optimum value which in this case, according to (13), is equal to  $(48 \cdot 0.7 \cdot R^3/3)^{1/4} \approx 15$  pixels. This value corresponds to a chord length of 30 pixels and is rarely possible to choose for a curve of radius 16. Therefore even a precision of  $\pm 38\%$  may be hardly reached in the case of  $\epsilon = 0.7$  and  $R = 16$ .

Thus, we may conclude that if  $\epsilon$  is about 0.7 pixel a precision higher than 10% may be only reached for long curves (about 250 pixels) of great radius (at least 270 pixels) which corresponds to a circular arc of about  $50^\circ$ .

Ways to increase the precision consist either in increasing  $dx$ , which is not always possible because of limited size of the curve segment whose curvature must be estimated, or in decreasing the value of  $\epsilon$ . As already mentioned in the Introduction, some published methods



of estimating the curvature are based on averaging the coordinates of subsequent points of the digital curve. If the digitization errors in the coordinates may be considered as random values then the RMS error of the coordinates may be reduced by the factor of  $\sqrt{N}$  in the case of averaging  $N$  points. This is true for averaging without weights. Otherwise the reduction is smaller. As we have seen in Section 1, this way yields a rather modest improvement since  $N$  is mostly a small number. Another method of essentially increasing the precision of the coordinates is suggested in the next section.

#### **4 The sub-pixel boundary and its detection method**

The estimation of the curvature is mostly being performed in binary images obtained by binarizing gray level images. The latter are produced either by CCD cameras or by scanners. Sometimes a scanner produces a binary image, however, this means that the binarization is performed inside the scanner.

There are cases when an image of a high contrast is being digitized, e.g. a technical drawing, or a map, or a machine part. In these cases the original objects of interest possess a sharp boundary whose location is specified in the original analog image with a relatively high precision. During the digitization and the following binarization a great deal of this information is lost. Therefore it is reasonable to try to get this information before the binarization, i.e. from the gray level image.

To solve this problem let us look, how a gray level image is produced in the case of an object having a sharp boundary. The object is being projected onto a set of light sensitive elements, e.g. CCD cells. Each cell is partially covered by the projection of the object. The amount of light received by a cell is a function of the area of the intersection of the projection with the light sensitive area of the cell. The amount of light is encoded as the gray level of a pixel corresponding to the cell.

If we know the function specifying the dependence of the encoded gray level from the said area of intersection and we suppose the boundary of the object to be a smooth curve with a limited curvature, then we can try to reconstruct the exact location of the boundary from the gray levels of adjacent pixels.

Let us consider the simplest model relevant to this problem. Consider a light object against a dark background and a rectangular grid of light sensitive cells. Let the light sensitive area of each cell be a square. Let us take the length of the square's side as the length unit. Thus the area of a cell is equal to 1. Suppose, there is no empty space between adjacent cells (or the space is negligible) and the cells do not overlap. Suppose that the curvature of the boundary of the object is so small, that the part of the boundary inside a cell may be considered as a straight line. We understand well that this model is too primitive to exactly describe the properties of the light sensitive elements of a scanner or a CCD camera. However, the model works, as we shall see in what follows.

Let us consider pairs of adjacent cells one of which has the intersection area with the object's projection less than 0.5 and the other one greater or equal to 0.5. Denote these two intersection areas by  $SA$  (small area) and  $GA$  (great area). The cells correspond to two adjacent pixels of the digitized image, composing a *pair*. We consider the pixels as squares of unit area in a plane which is a copy of the plane containing the idealized light sensitive cells with the projection of the object on it (Fig. 2; the light object is represented here and in Fig. 3 as a shaded area to make the figures nicer).

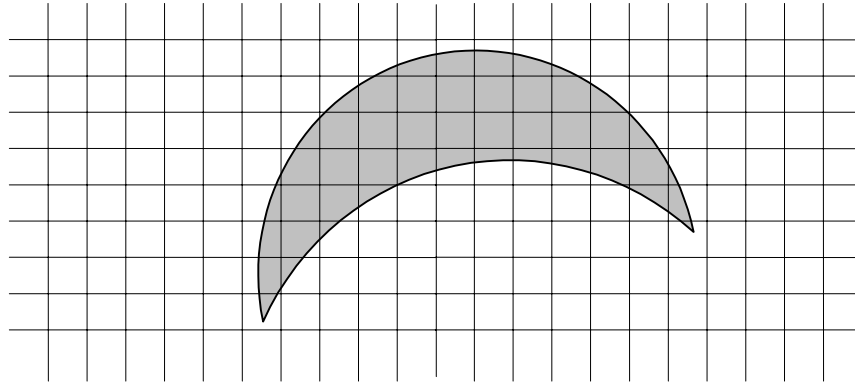


Fig. 2. An array of pixels with the projection of an object on it

The areas of the intersection of the pixels of a pair with the projection of the object are obviously equal to  $SA$  and  $GA$  correspondingly. If the gamma characteristic of the scanner is linear and the background has a zero brightness then the gray values of the pixels of a pair are proportional to the intersection areas  $SA$  and  $GA$ . Since an image produced by a scanner or by a CCD camera may contain some shading we normalize the said gray values:

$$SA = (GV_s - GV_{\min}) / (GV_{\max} - GV_{\min});$$

$$GA = (GV_g - GV_{\min}) / (GV_{\max} - GV_{\min});$$

where  $GV_s$  and  $GV_g$  are the gray values of the pixels of a pair,  $GV_{\min}$  and  $GV_{\max}$  are the minimum and the maximum gray values in a  $7 \times 7$  window around the pair.

Let us specify a gray value threshold lying in the middle between the minimum and the maximum gray values in the image. Then the gray value of the first pixel is below the threshold and that of the other one is greater or equal to the threshold.

It is easy to see that under the said conditions the boundary (which we suppose to be approximately a straight line inside a pixel) always crosses the segment  $S$  between the middle points of the pixels of a pair. Let us denote the crossing point having sub-pixel coordinates by  $SP$  and the distance from  $SP$  to the middle point of the pixel with  $GA$  by  $t$ . The values of  $SA$  and  $GA$  depend on  $t$  and on the angle  $\alpha$  between the tangent to the boundary and  $S$ .

There are 6 possible cases differing by the numbers of corners of the polygons corresponding to  $SA$  and  $GA$  as shown in Fig. 3.

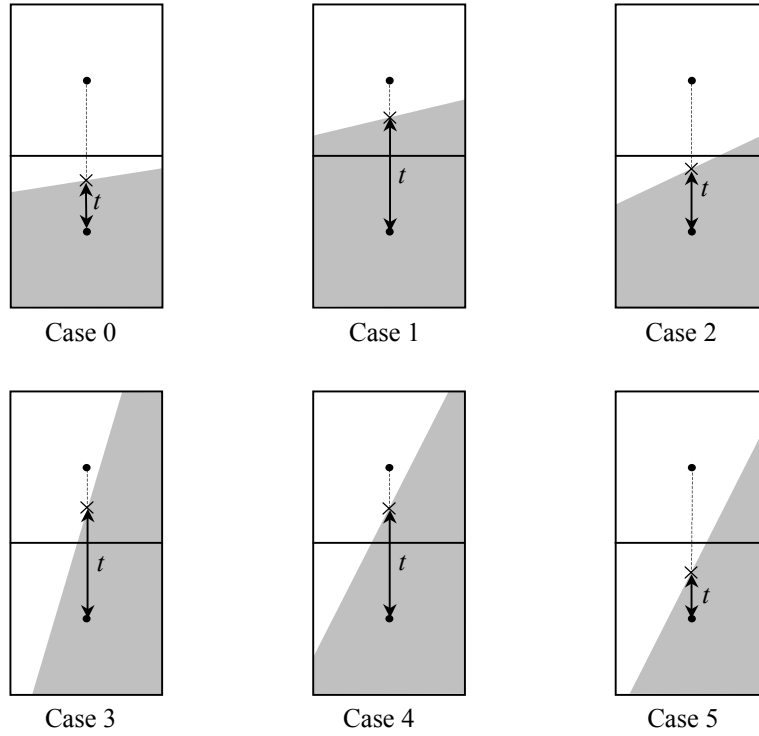


Fig. 3. The six possible cases of the intersections;  
the  $SP$ 's are marked by crosses, the length of the arrow is the parameter  $t$

In each case the intersection areas  $SA$  and  $GA$  may be specified as different functions of  $t$  and  $\alpha$ :

$$\begin{aligned}
 \text{Case 0: } & SA=0; \text{ and } GA=t+0.5; \\
 \text{Case 1: } & SA=t-0.5; \text{ and } GA=1.0; \\
 \text{Case 2: } & SA=0.5 \cdot L^2 \cdot \cot \alpha \text{ and } GA=1.0-0.5 \cdot K^2 \cdot \cot \alpha; \\
 & \text{with } L=0.5+(t-0.5) \cdot \tan \alpha \text{ and } K=0.5-(t-0.5) \cdot \tan \alpha; \\
 \text{Case 3: } & SA=0.5-(1.0-t) \cdot \tan \alpha; \text{ and } GA=0.5+t \cdot \tan \alpha; \\
 \text{Case 4: } & SA=0.5-(1.0-t) \cdot \tan \alpha \text{ and } GA=1.0-0.5 \cdot K^2 \cdot \cot \alpha; \\
 & \text{with } K=0.5-(t-0.5) \cdot \tan \alpha; \\
 \text{Case 5: } & SA=0.5 \cdot L^2 \cdot \cot \alpha \text{ and } GA=0.5+t \cdot \tan \alpha; \\
 & \text{with } L=0.5+(t-0.5) \cdot \tan \alpha.
 \end{aligned} \tag{18}$$

In each case we have a system of two equations in two unknowns  $t$  and  $\alpha$ . We have solved the equations and thus specified  $t$  and  $\alpha$  as functions of  $SA$  and  $GA$ . However, it is not enough: it is necessary to recognize, given  $SA$  and  $GA$ , which of the cases takes place. The solution of this problem was more complicated than the solution of the equations. It was found with the aid of a computer simulation. Nevertheless, the solution is an analytical one:

If  $GA=1$  then case 1;  
 else if  $SA=0$  then case 0;  
   else if  $SA > \max(GA/3, 3 \cdot (GA-3/4)+1/4)$  then case 3;  
     else if  $SA \geq 4 \cdot (GA-0.5)^2$  and  $GA < 3/4$  then case 5;  
       else if  $SA > 0.5 - 0.5 \cdot \sqrt{1-GA}$  and  $GA > 3/4$  then case 4;  
       else case 2.

This rather complicated relations are illustrated by Fig. 4.

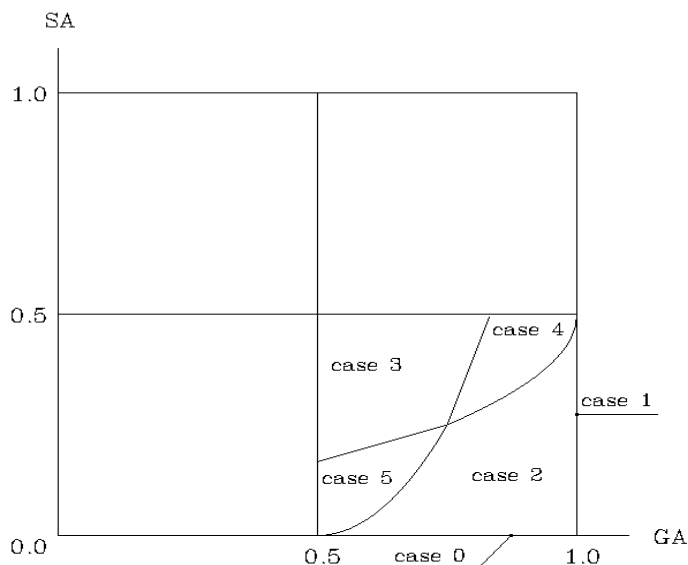


Fig. 4. Recognition of the six cases of dependencies between the intersection areas and the parameters  $t$  and  $\alpha$

The solution of equations (18) together with the recognition algorithm (19) yields for each pixel pair a sub-pixel point  $SP$  lying in the segment  $S$  between the middle points of the pixels. To find the pixel pairs it suffices to implicitly binarize the given gray value image with the threshold as specified above and to trace the crack boundary as described e.g. in [Kov92]. Each boundary crack found during the tracing separates two pixels composing a pair. Thus we can find for each boundary crack a *corresponding sub-pixel point*  $SP$  as shown in Fig. 5.

**Definition SPB:** The *sub-pixel boundary* of a region  $R$  in a gray value image (without shading) is the polygon whose vertices are the sub-pixel points and whose edges connect every two sub-pixel points corresponding to two adjacent boundary cracks of the usual boundary of the binarization of  $R$  with the threshold lying in the middle between the minimum and maximum gray values in the vicinity of the boundary of  $R$ .

The solution of (18) consists of six different pairs of analytical formulas: one for  $\alpha$  and one for  $t$  for each of the six cases. We have proved that the solution is continuous on the boundaries of the regions corresponding to the six cases in the plane  $(GA, SA)$  shown in Fig. 4. However, it is not continuous at the point  $GA=SA=0.5$ . At this point the equations are ill-conditioned with respect to  $t$ . We shall consider this case in the next section.

## 5 Estimating the curvature of the sub-pixel boundary

An estimate of the curvature may be computed for each boundary crack  $C_i$  by finding the circle running through the sub-pixel point  $SP_i$  assigned to  $C_i$  and through two other points  $SP_{i-d}$  and  $SP_{i+d}$  assigned to  $C_{i-d}$  and  $C_{i+d}$  correspondingly where  $d$  is a parameter specifying the half length of the chord. It is easy to see that the parameter  $t$  is ill-conditioned when  $\alpha$  tends to zero: a small inaccuracy in the values of  $SA$  and  $GA$  may lead to a great error in  $t$ . The situation takes place when a boundary crack is almost perpendicular to the tangent of the boundary of the preimage region  $R$ . (Fig. 6).

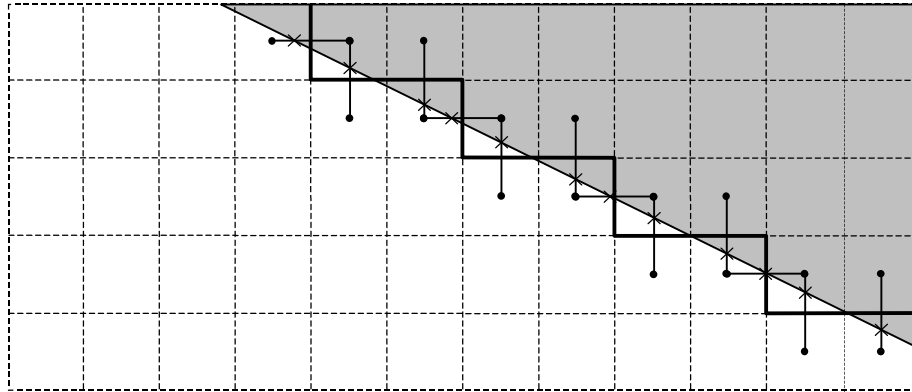


Fig. 5. An example of a boundary with sub-pixel points marked by crosses

In such cases it is better to discard the sub-pixel point corresponding to such a "bad" crack: its presence may only disturb the estimate of the curvature.

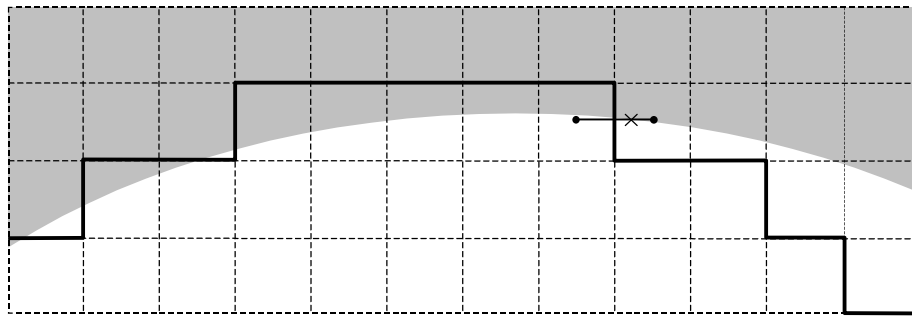


Fig. 6. An example of a boundary crack almost perpendicular to the tangent of the preimage boundary; the  $SP$  marked by a cross is ill-conditioned

Fortunately, there are few "bad" cracks in an image. Really, if the preimage boundary is a smooth curve with a restricted curvature then the "bad" cracks are singular. This means, there are at least  $N$  "good" cracks following a "bad" one, where  $N = \text{floor}(\cot \alpha)$  ( $\text{floor}(x)$  is the greatest integer less or equal to  $x$ ) and  $\alpha$  is as before the angle between the tangent to the boundary and the segment  $S$  connecting the middle points of the pixels of a pair. In the case of a "bad" crack,  $\alpha$  is small and  $\cot \alpha$  is large. Thus we may improve the precision of the curvature estimate while discarding a small number of boundary cracks.

Important is the choice of the parameter  $d$  defining the distance between the points in the triple  $(SP_{i-d}, SP_i, SP_{i+d})$  specifying the curvature circle. Here we have again, as in the case of estimating the first derivative of a function (Section 2), to do with an uncertainty similar to that by Heisenberg: the greater  $d$  the higher the precision of the curvature estimate, however, the greater the segment of the curve for which the curvature will be averaged and, therefore, the smaller the precision of estimating the location of the measurement.

If the curve under consideration is known to have a continuous curvature then the value of  $d$  may be chosen relatively great, e.g. 4 to 20 pixels. We suggest to choose the value of  $d$  automatically, as described below. The program first calculates the points of the sub-pixel boundary and stores them in an array. Then the program averages the coordinates of the points by calculating the convolution with a Gaussian kernel with the standard deviation  $\sigma$  of about 2 or 4 pixels. It calculates at first the curvature at each point with a relatively great value of  $d$ , e.g. equal to 1/8 of the number of cracks in the boundary of the object. Then the program reads the array of the sub-pixel points again, estimates coarsely the second derivative of the curvature with respect to the arc length (as an estimate of  $F4$  in (12)) and calculates the optimal value of  $d$  for each point. Then the program calculates the curvature with the optimal value of  $d$ . The results are demonstrated in Fig. 7 to 10.

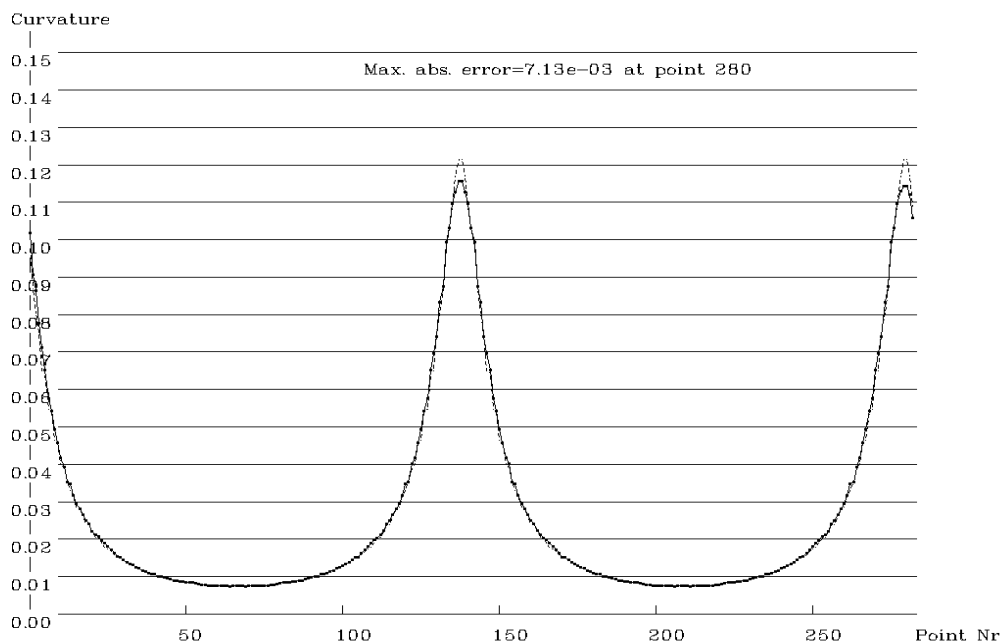


Fig. 7. The curvature of an artificial image of an ellipse (see text);  
the maximum relative error is about 6%

Fig. 7 shows the calculated curvature of the sub-pixel boundary of an artificial image of an ellipse with the half-axes of 53.11 and 20.86 pixels. The gray values of this image were calculated as being proportional to the intersection area of the ellipse with the corresponding pixel (regarded as a unit square). The gray values were rounded to integers in the range between 0 and 255 and saved each in one byte. The calculated curvature is represented by a

solid line; the true curvature, calculated analytically, is shown by a dashed line (which is only visible at the peaks). The errors in this case are only due to rounding the intersection areas and to the approximation of the parts of the ellipse inside a pixel by the tangent. The maximum error is about 6% of the maximum curvature.

Note, that it is not expedient to characterize the accuracy of a method of measurement by the maximum relative error since in the case when the result is near zero the relative error may become arbitrarily great. We prefer to use the relation of the maximum absolute error to the length of the interval in which the results produced by the method with fixed parameters lie.

Fig. 8 shows the curvature of the same ellipse represented as an image printed on a sheet of paper with a laser printer with a resolution of 300 DPI.

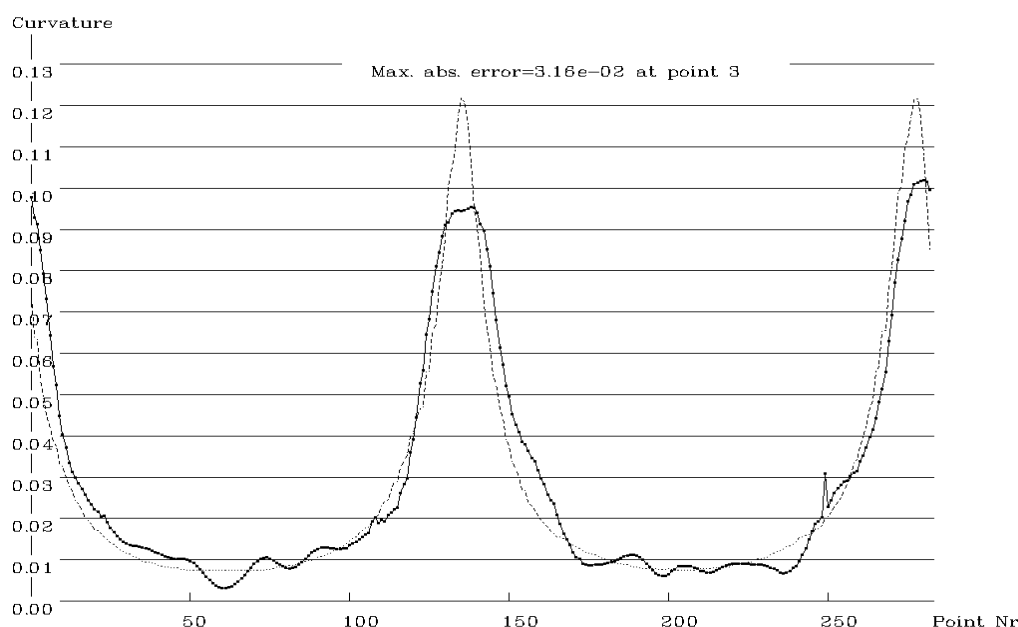


Fig. 8. The curvature of a printed and scanned image of an ellipse (see text); the dashed line shows the true curvature; the maximum relative error is about 26%

The image was scanned with a resolution of 25 DPI to make the pixel size essentially greater than the irregularities of the boundary due to the limited resolution of the printer. The errors are essentially greater than in the case of Fig. 7 although the parameters of the program and of the ellipse were the same. The cause is the irregularity of the boundary in the printed image: such an image is not a true analog image.

The results are much better when scanning a photograph or a real object like that shown in Fig. 9. It represents a link of a bicycle chain. The object is 21.0 mm long and 8.25 mm wide. The curvature radii of the outer boundary are equal to 4.125 mm at the left and right side and to -6.6 mm in the middle. The image was scanned by a digital camera with a resolution of

155 DPI. Thus the radii in pixels are equal to 25.19 and  $-40.32$  pixels respectively, which corresponds to curvatures of  $0.0397$  and  $-0.0248$   $\text{pixel}^{-1}$ .



Fig. 9. The gray value image of a real object used in the experiments

Fig. 10 shows the curvature of the outer boundary of the object of Fig. 9 calculated by our method. The errors in this case are only due to rounding of the intersection areas during the scanning and to the limited precision of the method.

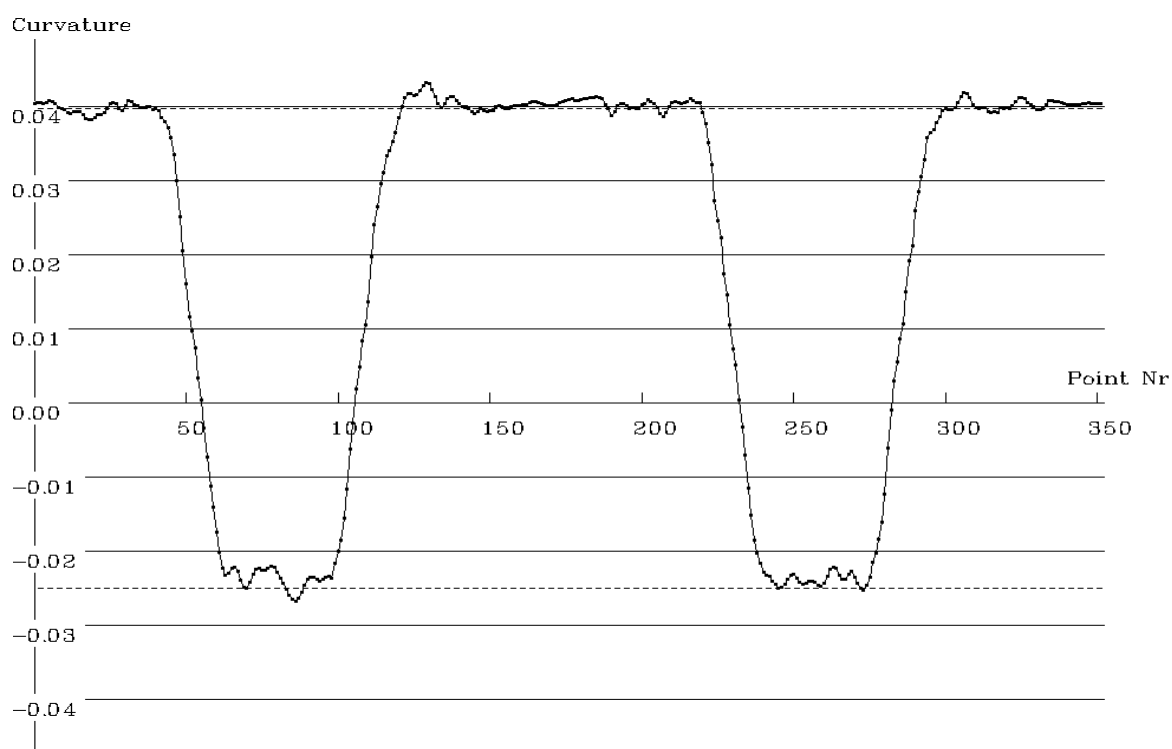


Fig. 10. The curvature of the outer boundary of the object shown in Fig. 9; the dashed lines show the true values of the minimum and maximum curvature

It was impossible to calculate the error at each point of the boundary, as we have done in the case of the ellipse, since we don't know the exact shape of the object. We only could measure the curvature radii at the points of extrema. In the neighborhood of these points the maximum absolute error is about  $(0.0220 - 0.0248) = -0.0028$   $\text{pixel}^{-1}$ . Being divided by the interval length



of  $0.0397 - (-0.0248) = 0.0645$  it results in the relative error of about 4%. We hope that Fig. 10 gives a general impression of the accuracy of the method as applied to real objects.

## 6 Conclusion and Discussions

We have shown that the naive belief, that to estimate the derivative of a numerical function  $y=f(x)$  one should calculate the value of  $dy/dx$  with as small value of  $dx$  as possible, is wrong. It may lead to great errors. We also have shown how to choose the value of  $dx$  optimally.

The same considerations being applied to the calculation of the second derivative enable one to find the optimal distance between the three points on a curve, which distance minimizes the maximal possible error in estimating the curvature of the curve.

We have suggested a new method of defining and calculating the sub-pixel boundary of a region in a gray value image and we have developed a new method of estimating the curvature of boundaries of regions in gray value images. It was demonstrated that the method being applied to a gray value image obtained by scanning a scene containing a dark object on a light background (or vice versa) yields an essentially higher precision of estimating the curvature as the known methods.

However, there are cases, in which the boundary of a region in the preimage is a continuous curve consisting of some smooth pieces while the tangent to the curve is not continuous at the border of a piece. In such cases the digital curve must be segmented into pieces whose preimages are supposed to have a continuous tangent. We see two possibilities to solve the segmentation problem. One of them is based on the polygonal approximation (or subdivision into digital straight line segments). The other solution is based on the recognition of digital circular arcs [Kov90]. According to this method a digital curve must be traced, and its segment between the starting and the running point must be tested, whether it may be considered as the result of the digitization of a continuous circular arc. When this condition is no more fulfilled, the part of the digital curve between the starting point and the point before the last one is a digital circular arc (DCA). The curvature at all points of the DCA can be estimated by the sub-pixel method as described above, without crossing the end points of the DCA. The suggested methods of segmentation may improve the estimates of the curvature at locations where the curvature of the preimage has a jump, e.g. at a location where two circular arcs of different radii have a common tangent. The investigation of the methods of segmentation is, however, a topic for a separate publication.

The suggested approach may also be applied to estimate the curvature of a surface in the 3D space if the surface is given as the boundary of a connected set of voxels while each voxel contains a non-binary density value as this is the case e.g. in computer tomography. It is well-

known from differential geometry that the principal curvatures and their directions are solutions of quadratic equations whose coefficients may be expressed through the first and second partial derivatives of the radius-vector with respect to the parameters of the parametric equation of the surface. The derivatives may be estimated on the basis of the densities in adjacent voxels in a way similar to that described in Section 4.

## References

- [Bu92] Buchanan, James L., and Turner, Peter R., *Numerical Methods and Analysis*, McGraw-Hill 1992, pp. 422-424.
- [Hla94] Hlavác, V., Pajdla, T. and Sommer, M., *Improvement of the Curvature Computation*, In: Proceedings of the 12<sup>th</sup> International Conference on Pattern Recognition, IEEE Press, vol. 1, 1994, pp. 536-538
- [Kov90] Kovalevsky, V., *New Definition and Fast Recognition of Digital Straight Segments and Arcs*, In: Proceedings of the 10<sup>th</sup> International Conference on Pattern Recognition, IEEE Press, vol. 2, 1990, pp. 31-34
- [Kov92] Kovalevsky, V., *Finite Topology and Image Analysis*, Chapter in "Image Mathematics and Image Processing", P. Hawkes (Ed.), Series "Advances in Electronics and Electron Physics", Academic Press, vol. 84, 1992, pp. 197-259
- [Low89] Lowe, D.G., *Organization of Smooth Image Curves at Multiple Scales*, International Journal of Computer Vision, vol. 3, 1989, pp. 119-130
- [Via96] Vialard, A., *Geometric Parameter Extraction from Digital Paths*, In: Miguet, S., Montanvert, A., and Ubéda, S. (Eds.), "Discrete Geometry for Computer Imagery", Springer, pp. 24-35, 1996
- [Wor93] Worring, M., *Shape Analysis of Digital Curves*, Doctor Thesis. University of Amsterdam, 1993
- [WS93] Worring, M. and Smeulders, A.W.M., *Digital Curvature Estimation*, CVGIP: Image Understanding, v. 58, No 3, 1993, pp. 366-382
- [DF90] Deriche, R., and Faugeras, O., *2-D Curve Matching Using High Curvature Points: Applications to Stereo Vision*. In: Proceedings of the 10<sup>th</sup> International Conference on Pattern Recognition, IEEE Press, vol. 1, 1990, pp. 240-242