

# Estimation of the curvature of an interface from a digital 2D image

Olav Inge Frette<sup>a,b,\*</sup>, George Virnovsky<sup>a</sup>, Dmitriy Silin<sup>c</sup>

<sup>a</sup>*International Research Institute of Stavanger (IRIS), Prof. Olav Hanssensvei 15,  
NO-4068 Stavanger, Norway*

<sup>b</sup>*Department of petroleum technology, University of Stavanger, NO-4036  
Stavanger, Norway*

<sup>c</sup>*Lawrence Berkeley National Laboratory, 1 Cyclotron Road, MS 90-1116,  
Berkeley, CA 94729, USA; and University of California, Berkeley, 415 Davis Hall,  
Berkeley, CA 94720, USA*

---

## Abstract

In this paper a method for the estimation of the curvature along a condensed phase interface is presented. In a previous paper in this journal [1] a mathematical relationship was established between this curvature and a template disk located at a given point along the interface. The portion of the computed area of the template disk covering one of the phases was shown to be asymptotically linear in the mean curvature. Instead of utilizing this relationship, an empirical approach was proposed in [1] in order to compensate for discrete uncertainties. In this paper, we show that this linear relationship can be used directly along the interface avoiding the empirical approach proposed earlier. Modifications of the algorithm are however needed, and with good data smoothing techniques, our method provides good quantitative curvature estimates.

*Key words:* Curvature estimation, condensed phase interface, digital image analysis, porous rock, curvature-driven processes  
*PACS:* 02.40.Hw, 06.30.Bp, 07.05.Pj, 61.43.Bn, 61.43.Gt, 68.35.Ct

---

## 1 Introduction

Capillary phenomena can have an important role in different parts of materials science [2]. With capillary phenomena we mean processes involving surfaces<sup>1</sup> which occur on length scales where the curvatures play an important role. Over the past decades the study of high-temperature capillarity has developed as a distinct research field with contributions both to fundamental research and applications; see the recent special issue of *Current Opinion in Solid State and Materials Science* on this diverse subject [3]. Curvature-driven processes include grain boundary migration in bicrystals [4], dendritic growth in solidification of alloys [5], dissolution driven crack growth [6], surface diffusion in molecular beam epitaxy [7] and undercooling in eutectics [8]. Processes like these include the movement or evolution of surfaces. Different approaches to computer modeling exist for the simulation of these movements like level set [9], fast marching [9], cellular automaton [10,11], diffusion-limited aggregation [12], and equivalent sharp surface [13]. For simulations performed on digital images [14], the curvature needs to be computed along the boundary pixels of the moving surface.

---

\* Corresponding author. Tel.: +47 51875026; fax: +47 51875200.

*Email address:* Inge.Frette@iris.no (Olav Inge Frette).

<sup>1</sup> We are aware of the difference in the meanings of surface and interface in the literature, but for our purpose these concepts are synonymous.

In this paper we present an algorithm for computation of the curvature of a smooth curve from its digital image. This algorithm is a modification of an algorithm published in [1] some years back. The basic idea is as follows. Given a curve in 2D, place a template disk of a given radius with its center at a point along the curve. Asymptotically, the curvature of the curve at this point dependent linearly on the area of the template disk that is on one side of the curve. So the curvature can be estimated by computing this area.

Application of this algorithm to a digital image depends on the resolution, which determines the discretization of the surface. For our purpose, a modification of the established algorithm was needed, and this new modified approach is presented in this paper. It should be noted that this algorithm is suitable not only for material science simulations of surface evolution. It can just as well be used in other scientific and engineering disciplines where digital pictures are used and where quantitative estimates of the curvature is needed. Some examples are heart wall analysis (medical science) [15], diatom identification (biological science) [16], shape perception (robot engineering) [17] and thin film stability (petroleum engineering) [18].

Our paper has three parts - theory, method and results. In the theoretical part a short introduction into the mathematics of curvature is followed by an overview of common approaches to the estimation of the curvature of a curve from its discretized image. Then we present our new approach for curvature estimation, followed by a comparison with the algorithm in [1]. In the method part we describe our algorithm. In the last section we present results of computations with this algorithm and compare them to analytical solutions. The algorithm has also been applied to analyze real data – SEM images of sedimentary rocks. Results of computed curvatures along the wall of a pore in the

porous rock are included.

## 2 Theory

### 2.1 The curvature of a planar curve: continuous case

We will use the following formula for the curvature  $\kappa$  of a 2D curve  $x = x(t)$ ,  $y = y(t)$  parameterized by a parameter  $t$  [19,20]:

$$\kappa = \frac{y''x' - y'x''}{[x'^2 + y'^2]^{3/2}} \quad (1)$$

If  $\theta$  is the angle between the tangent line and the x-axis and the arc length  $s$  is selected as the parameter  $t$ , then, equivalently,

$$\kappa = \frac{\theta'}{\sqrt{x'^2 + y'^2}} = \frac{d\theta}{ds} \quad (2)$$

We will also employ two more equivalent definitions of curvature

- Path based definition:  $\kappa = \pm \|\mathbf{X}''(s)\|$ , where  $\mathbf{X}(s) = (x(s), y(s))$  and  $s$  is the length of the curve.
- Osculating circle based definition:  $\kappa = \pm \frac{1}{R(s)}$ , where  $R(s)$  is the radius of the osculating circle touching the curve.

In both cases, the choice between “+” and “-” is determined by the local convexity convention.

Numerical evaluation of the curvature of a curve using any of the formulae presented above require a numerical differentiation, which is an ill-posed problem [21]. Below, we present an alternative approach.

## 2.2 *Computing the discrete curvature of a curve*

Several methods can be proposed to evaluate numerically the curvature of a discrete object. For example, one could map the discrete object into continuous space by using methods like orthogonal polynomials, linear regression, or B-splines [22], and then calculate the curvature in a point using one of the definitions for continuous curvature presented in the previous subsection. Several problems associated with such a method have been noticed. Firstly, the methods are limited since the input area needs to be reduced to provide a given accuracy. Secondly, the algorithms based on methods like these are time consuming for large input data [23].

In a widely referenced paper by Worring and Smeulders [24], different approaches found in the literature for the computation of discrete curvatures are evaluated. The approaches are divided into three groups corresponding to different formulae for the curvature, from the previous subsection: the orientation-based approach which uses equation (2), the path-based approach which according to Worring and Smeulders also include equation (1), and the osculating-circle based approach. They note that these definitions are equivalent in a continuous space, but not necessarily so in a discrete one.

For the orientation-based approaches three methods are evaluated. In the first method, the curvature is estimated using a low pass differential filtering of the estimated tangent direction. The second method is similar to the first one except for resampling of the discrete contour in order to reduce the anisotropy of the grid. The third approach is a line fit method. The curvature is calculated finding the angular difference between two straight lines fitted to the data near

the point of consideration. For the path-based approach one method is evaluated. Using differential Gaussian kernels on the  $x$ - and  $y$ -coordinate sequences the curvature can be estimated. In the osculating circle-based approach, the curvature can be obtained by finding the circular disk that fits the contour points best. Further literature references regarding the evaluation of different methods can be found in Worring and Smeulders' paper [24].

### 2.3 Computing curvature using a template disk

Our approach for computing the curvature at a point along a boundary of a discrete object is based on the idea of computing the partial area of a template disk that is placed with its center in the given point. Since there exists an asymptotically linear relationship between this computed partial area and the curvature, the curvature value can be estimated.

The theoretical basis for this approach is as follows. For a given curve separating two 2D "phases" a disk template is placed with its center at a point  $P$  of the curve, see figure 1. The radius  $b$  of the template disk is assumed to be small compared to the radius of curvature of the curve at  $P$ . Assuming sufficient smoothness of the curve, we locate the 2D Cartesian coordinate system with origin in  $P$  and the  $x$ -coordinate axis is placed along the tangent line of the curve. The position and orientation of the coordinate system imply  $f(x)_{x=0} = \left(\frac{df}{dx}\right)_{x=0} = 0$ . Hence,

$$y = f(x) = \frac{1}{2} \left(\frac{d^2f}{dx^2}\right)_{x=0} x^2 + O(x^3) \quad (3)$$

By virtue of Equation (1),

$$y = -\frac{1}{2}\kappa x^2 + O(x^3) \quad (4)$$

Rewriting this equation in dimensionless form scaled with the radius  $b$  of the template disk, ( $X = x/b$ ,  $Y = y/b$ ,  $K = \kappa b$ ), yields

$$Y = -\frac{1}{2}KX^2 + O(X^3) \quad (5)$$

In polar coordinates  $r$  and  $\theta$ , with  $R = r/b$ , the equation can be written as

$$R \sin \theta = -\frac{1}{2}KR^2 \cos^2 \theta + O(R^3) \quad (6)$$

Solving for  $\theta$ , we get

$$\theta(R) = \sin^{-1} \left[ -\frac{1}{2}KR + O(R^3) \right] \quad (7)$$

The angle  $\theta$  can now be interpreted as the angle between the tangent line passing through  $P$  and the  $x$ -axis. The area  $A$  of the template disk outside the line segment can then easily be calculated (in scaled form  $a = Ab^{-2}$ ) as

$$a = \int_0^1 R dR \int_{\theta(R)}^{\pi - \theta(R)} d\theta \quad (8)$$

Substitution of Equation (7) into (8) yields

$$a = \frac{\pi}{2} - \frac{1}{3}K + O(R^3) \quad (9)$$

Or, in terms of  $K$ ,

$$K = 3a - \frac{3\pi}{2} + O(R^3) \quad (10)$$

The error in the last equation is small compared to the true curvature. Thus equation (10) can be written (in physical coordinates)

$$\kappa \simeq \frac{3A}{b^3} - \frac{3\pi}{2b} \quad (11)$$

The above equation (11) establishes the linear relationship between the curvature  $\kappa$  and the area  $A$ . This functional relationship will be the basis for further discussion of curvature estimation below.

#### 2.4 *Computing curvature using a discrete template disk*

The method described above can be applied to digital images or grids. In either case it is assumed that the image or grid is represented in a binary form by being divided into a *background* and into one or more *regions*. By convention, the background is assumed to be white while the regions are assumed black. The points in the image or grid are represented as square pixels. The discrete template disk is placed with its center along the boundary pixels of the regions. The calculation of area  $A$  then reduces to the counting of pixels belonging to that part of the template disk laying outside the regions. Discretization inevitably introduces errors in the curvature evaluation.

Firstly, there is an error in the area of the discrete template disk compared to the theoretical template disk. This error depends on the diameter of the disk; it decreases when the diameter of the disk increases, see figure 2. On the other hand, equation (11) requires that the radius of the disk is small compared to the radius of curvature. Hence the optimal size of the template disk should be chosen depending on the curvature of the boundary. At the same time large diameters contribute to an additional error due to non-uniformity of the

curvature over larger area of the boundary.

When using equation (11) in a discrete way an additional error is introduced since the area  $A$  is a discrete parameter while the radius  $b$  is not. Thus equation (11) needs to be reformulated in terms of fractional area. By introducing  $A_{tot} = \pi b^2$ , equation (11) can be written as

$$\kappa \simeq \frac{3\pi}{b} \left( \frac{A}{A_{tot}} - \frac{1}{2} \right) \quad (12)$$

$A_{tot}$  is the area of an analytical disk with radius  $b$ .

By approximating  $b$  from  $b^2 \simeq A_{tot}^d/\pi$  (where  $A_{tot}^d$  is the total area of the discrete template disk) and applying this expression to equation (11) we get

$$\kappa \simeq \frac{3\pi}{b} \left( \frac{A}{A_{tot}^d} - \frac{1}{2} \right) \quad (13)$$

Since both the numerator and the denominator are counted in pixels in equation (13), the fractional area,  $\frac{A}{A_{tot}^d}$ , is calculated in a consistent manner. Equation (13) establishes the linear relationship between the computed area and curvature that will be used in this paper.

### 3 Our Method

Our approach focus on the application of equation (13) directly to the digital image.

### 3.1 *Boundary representation*

There are basically two different ways of representing the boundary (also called border, contour) of such a region. The first is a pixel-based approach [17,25]. In this approach, the boundary is represented by those pixels in the regions that are neighbors to background pixels; see figure 3. If the pixels in the regions are 4-connected all boundary pixels must contain at least one 8-connected pixel belonging to the background. If the regions are 8-connected all boundary pixels must contain at least one 4-connected pixel belonging to the background [25]. In [1] the boundaries are pixel-based, and the boundary pixels are called surface pixels. It is not clear whether the regions are 4- or 8-connected.

The second way of representing the boundary is the edge-based approach; also called the inter-pixel approach [26,27]. Every square pixel has four sides - left, right, top and bottom. These sides are also called edges. In this approach, the boundary of a region is understood as the collection of edges that are shared between the region pixels and background pixels; i.e. shared between complementary domains, see figure 4. These edges are called crack edges. The two approaches to boundary representation have different properties. Pixel-based boundaries are dependent on the definition of the neighborhood, the area of the boundary is greater than zero and the boundaries are not shared with adjacent regions. On the other hand, edge-based boundaries have zero area, are shared with adjacent regions, but contain sub-pixel elements - the edges. In our paper we use the edge-based approach, and we think that this representation of the boundary is well suited for the curvature algorithm presented below.

### 3.2 *The discrete template disk*

We use template disks with odd pixel diameters, like in [1]. A discrete template disk with a given diameter is generated by including all pixels which have their center located at a distance from the center of the circle which is less than  $b$ , the radius of the circle. Radius  $b$  is defined as  $2b = N\xi$  where  $N$  is an odd number and  $\xi$  is defined as one pixel unit. Figure 5 shows an example of a template disk with pixel diameter equal to 15.

### 3.3 *The simulation approach*

For a given digital image that has been segmented, the curvature is computed for each region. This is done by moving the template disk along the region boundaries with its center in the boundary pixels, see figure 5. The contributing area of the disk will be calculated and from the curvature can be computed from equation (13). It turns out that this approach still needs some modification in order to get good results. Finally some data smoothing removes high frequency noise from the computed values.

## 4 **Algorithm modification, verification, and results**

In this section, we present results for our algorithm. Firstly we describe the adjustments needed. Followed by this is a discussion about data smoothing where we propose a simple adaptive algorithm of reduction of the high-frequency noise. Thirdly the question about resolution is discussed. Then we compare our approach with the method of Bullard et al., followed by results from a

test using a SEM image of a sedimentary rock. Finally, we discuss some uncertainties and limitations inherent in our method.

#### *4.1 Test data set*

For testing and verification we have used a synthetic case as digital region - a digitized sinusoidal interface. This sinusoidal is defined as  $y = C \sin(kx)$  where  $C = 30\xi$ ,  $k = \frac{2\pi}{\lambda}$  with  $\lambda = 100\xi$ , see figure 6. As above,  $\xi$  is equal to one pixel unit. This test case was chosen since Bullard et al. used a similar one in their paper. For the discussion about resolution quality we also used two other sinusoidal interfaces - one which was twice the resolution of the original and one which was four times the resolution of the original. Results from an image of a sedimentary rock are included in 4.6.

#### *4.2 Algorithm modifications*

When computing the curvature using equation (13) and applying the template disk as described above, we noticed a bias in the computed values compared to the analytical results. The reason for this systematic error is related to the choice of boundary.

##### *4.2.1 Choice of boundary*

In their paper Bullard et al. used template disks with odd pixel diameters. With an odd diameter there is a unique center pixel in the disk. When the template disk is applied along a given boundary region, the disk is placed with its center pixel in each surface pixel, see figure 5. The number of pixels

in the template disk that is located outside the given region is counted and from this pixel number a curvature value is computed using equation (17). We have also used odd pixel diameter for the template disk. However, since we use edge-based boundary we need to find the corresponding boundary pixels (i.e. the surface pixels) in order to use the template disk properly. Basically we do the same as Bullard et al., but instead of using equation (17) the curvature is computed using equation (13) instead. Results from a simulation using a pixel diameter of 15 are shown in figure 7. A drift in the simulated results is observed. We also did a similar simulation where the template disk was applied to the so-called outer boundary<sup>2</sup>. Figure 7 shows results from this simulation. We notice that the results are symmetrical around the analytical curve. The errors are similar, but have opposite signs. This suggests that the solution must lie somewhere “in between” those two situations. The two simulated curves are computed from boundaries with extended thickness of one pixel. The edge-based boundary on the other hand has zero thickness, like an ideal analytical curve. Computing the curvature for a given boundary edge instead of a boundary pixel could possibly be an approach to avoid the reported drift errors. To accomplish this we need two computations for each boundary edge. The template disk will be placed with its center both in the inner and outer boundary pixels corresponding to the given edge, see figure 8.

The average value of the two computed areas is then used as input value for

---

<sup>2</sup> A pixel-based boundary of a given region is defined as all pixels of this region that are neighbors to background pixels. Alternatively, the boundary can be defined as the set of background pixels that are neighbors to the pixels of the given region. These two types of boundaries are respectively called inner and outer boundary [28].

$A$  in equation (13), which can be rewritten as

$$\kappa \simeq \frac{3\pi}{b} \left( \frac{A_{ave}}{A_{tot}^d} - \frac{1}{2} \right) \quad (14)$$

where

$$A_{ave} = \frac{A_{in} + A_{out}}{2} \quad (15)$$

$A_{in}$  is the number of pixels in the template disk covering the background when the center of the disk is located at the inner boundary. Similarly,  $A_{out}$  corresponds to the situation when the center of the disk is located at the outer boundary. Notice that one must use identical disk diameters for these two computations. Figure 9 shows the results of curvature computation with the adjustments included: the bias is practically eliminated.

### 4.3 Data smoothing

All the figures based on the sinusoidal simulations have shown smooth results for the curvature values. High frequency noise is unavoidable given the discrete nature of the region boundaries. Figure 10 presents an example. Both the original curvature plot and the smoothed version are included together with the analytical curve. We observe that the noise fluctuations are distributed along the whole curve. It seems that this noise is critical for small curvatures since it can fluctuate between positive and negative values. This could be significant, for example, when used in simulations of curvature driven processes. In these processes the sign of the curvature indicates the direction of the process. The low frequency signal is close to the analytical value. Thus it makes sense to apply some form of low pass filtering to remove the high frequency noise in

the curvature signal. There exist different methods; running averages, median and least square filtering. Fixed windows or adaptive windows can be used in conjunction with these methods. Fourier analysis can also be used: the noisy part of the signal can be removed by truncation of the data in the frequency space.

We propose a simple model for data smoothing using a running average algorithm with adaptive window size where the window size is estimated from the curvature value as

$$w = 0.5 * \frac{1}{|\kappa|} \tag{16}$$

$w$  is the window size in pixel units. All results presented have included data smoothing using this approach. We have used a lower bound for the window size of size 3 and size 20 for the upper bound.

#### 4.4 Resolution

We can expect that the algorithm presented in this paper will provide better results with a higher resolution. This has been tested by using two sinusoidal images at resolutions x2 and x4 defined relative to the original image, see Table 1. Curvature values have been computed for each case and rescaled for comparison with the original sinusoidal. In figure 11, a subpart of the results from the comparison is plotted. The average error of computed curvature - i.e the average value for the difference between computed value and analytical value - and the standard deviation has been estimated for all three sizes. This has been done both for the original computed values and for the smoothed values. The results show that a refinement of the resolution reduces the noise,

and this confirms that higher resolution gives more precise results. When data smoothing has been applied the average difference between computed value and analytical value is reduced by 45% for the image with 2x resolution and by 60% for the image with 4x resolution. For the standard deviation there is a reduction of 30% and 50% respectively.

#### 4.5 Comparison with the method of Bullard et al.

Our approach differs from that of Bullard et al. [1]. We established equation (14) using fractional area in order to avoid discrete error inherent in equation (11) and area measurements of both inner and outer boundaries to avoid error bias. For the computation of curvature we use equation (14) directly. Bullard et al. do not use equation (11) (which corresponds to equation (18) in [1]) directly either. Instead they suggest an “experimental” approach. They apply the discrete template disk on a set of discrete test circles with different radii. For each test circle they let the template disk move along the boundary and for each boundary pixel the outside area is computed by counting the number of pixels in the template disk not covering the test circle. Then an average value is estimated for the area. These average values are then plotted against the true curvature of the test circles. Using linear regression the following relationship (equation 22 in [1]) is established from these “experimental” tests for a template disk of 15 pixel diameter

$$\kappa = \frac{2.807}{b^3} \langle A \rangle - \frac{2.586\pi}{2b}, \quad (17)$$

where  $\langle A \rangle$  is the average pixel count.

Note that equation (17) is valid only for  $b=15$ . There may however be situations where it would be more useful to apply template disks with diameters different from 15. The computation of curvatures using images with very high resolution and smooth region boundaries is one example. Another one is computing the curvature of large-scale trends of the region boundaries. Finally we mention situations where the regions in the image are so small that template disks of 15 pixel diameters can not be used. For the two first examples an increased pixel diameter is needed, while for the last example a reduced pixel diameter is needed. When using the approach by Bullard et al. new numerical experiments would be needed for every pixel diameter of the template disk in order to establish the correct correlation between area and curvature, which is cumbersome. Equation (17) can not be used for other pixel diameters, as figure 12 illustrates, resulting in incorrect curvature values. It remains unclear what sampling size is sufficient for justification of the correlation-based equation (17). This size and the fitting coefficients may differ from curve to curve. Our approach does not rely on statistics. Rather, by direct computation of the curvature using equation (14), any error bias is dramatically reduced by symmetrization of the concept of the boundary.

#### *4.6 Computed curvature for the geometry of a pore in a sedimentary rock*

The algorithm has been applied to a digital image of a thin section of a sedimentary rock sample, see figure 13. After preprocessing, including image segmentation, a single pore was isolated for this test, figure 14. Computed curvature values for a subpart of the pore (enclosed by the grey rectangle) is presented in figure 15. The inner boundary of the pore is also included

in this figure. The width of the subpart is 160 pixels, and the height of the inner boundary pixels is given relative to the lowest point along the boundary. The unit of the curvature values are given in  $[\text{pixel}^{-1}]$ . To find the physical curvature the computed values need to be rescaled using the resolution of the image.

It is difficult to evaluate these results quantitatively since the true curvatures are unknown. Assuming a proper threshold value in the segmentation process, the results are qualitatively as expected. Both the sign and value of the computed curvatures are consistent with the shape of the pore boundary, and for flat segments of the boundary the curvature values are close to zero. Since it has been shown from the synthetic test case that this algorithm gives good qualitative results, we think that - within expected numerical errors - the computed curvature values for the realistic test case are reliable.

#### *4.7 Uncertainties and limitations*

Numerous factors and assumptions may affect the output of the algorithm. The impact of image resolution has been discussed above. Small variations of segmentation threshold can assign an originally grey-scale image pixel or a group of pixels either to the background or a region. The angularity of natural rocks cannot be completely resolved due to the finiteness of the resolution. Some features, which can be naturally present in rock images, like grain-to-grain contacts, make the very applicability of the concept of curvature questionable. In a forthcoming paper [18] we will discuss the possible effect the segmentation process has on the computation of the curvatures. Different sorts of angularities and contacts can be detected once a database

of digital signatures has been generated. This subject is beyond the scope of this study.

## **5 Summary and conclusions**

We have presented a modified algorithm for the computation of curvature in digital images or grids. The approach is based on an edge-based boundary which has the advantage of having no boundary thickness. The algorithm has been modified to avoid errors related to the understanding of boundaries. With a suitable data smoothing algorithm we are able to present computed values that are in good agreement with simulated values from a synthetic test case. We have also shown that an increase in resolution will reduce the noise in the results. With the memory and processing power available nowadays - even with a standard PC - these simulations are performed in a very short time span. Thus the inclusion of edges (which increases the memory use) and the double curvature computation for each edge (which increases the processing time) makes this approach attractive from a computational point of view. Using the algorithm on a realistic data set gives consistent and reliable results. The approach in [1] relies on empirically constant coefficients which could be inappropriate for different types of curves. The universality of our algorithm makes it applicable for any types of curves.

## **6 Acknowledgement**

The authors acknowledge financial support from the Norwegian Research Council and from ConocoPhillips and the Ekofisk Coventurers, including TO-

TAL, ENI, StatoilHydro and Petoro, through the research center COREC. This work was additionally supported in part by the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. The authors also thank Erich Suter for valuable discussions.

## References

- [1] J. W. Bullard, E. J. Garboczi, W. C. Carter, and E. R. Fuller Jr., Numerical methods for computing interfacial mean curvature, *Computational Materials Science*, 4, 103-116, 1995.
- [2] R. DeHoff, *Thermodynamics in materials science*, second ed., CRC, Boca Raton, 2006
- [3] Special issue on high temperature capillarity, *Current Opinion in Solid State and Materials Science*, 9, 2005, 149-253
- [4] A. D. Sheikh-Ali, The effect of grain boundary sliding on curvature-driven boundary migration in Zn bicrystals, *Scripta Materialia*, 56, 2007, 1043-1046.
- [5] M. F. Zhu, D. M. Stefanescu, Virtual front tracking model for the quantitative modeling of dendritic growth in solidification of alloys, *Acta Materialia*, 55, 2007, 1741-1755
- [6] P. Stahle, C. Bjerken, A. P. Jivkov, On dissolution driven crack growth, *International journal of Solids and Structures*, 44, 2007, 1880-1890
- [7] A. -L. Barabasi, H. E. Stanley, *Fractal concepts in surface growth*, Cambridge University Press, Cambridge, 1995
- [8] K. A. Jackson, *Kinetic processes - crystal growth, diffusion, and phase transitions in materials*, Wiley-VCH, Weinheim, 2004
- [9] J. A. Sethian, *Level set methods and fast marching methods - evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, Cambridge University Press, Cambridge, 1999

- [10] D. P. Bentz, P. J. P. Pimienta, E. J. Garboczi, W. C. Carter, Cellular automaton simulations of surface mass transport due to curvature gradients: simulations of sintering in 3-D, *Proc. Mater. Res. Soc.*, 249, 1991, 413
- [11] P. J. P. Pimienta, E. J. Garboczi, W. C. Carter, Cellular automaton algorithm for surface mass transport due to curvature gradients: simulations of sintering, *Computational Materials Science*, 1, 1992, 63
- [12] R. Tao, M. A. Novotny, K. Kaski, Diffusion-limited aggregation with surface tension, *Physical Review A*, 38, 1988, 1019-1026
- [13] J. W. Bullard, Digital-image-based models of two-dimensional microstructural evolution by surface diffusion and vapor transport, *Journal of Applied Physics*, 81, 1997, 159-168
- [14] R. Klette, A. Rosenfeld, *Digital geometry - geometric methods for digital picture analysis*, Elsevier, San Fransisco, 2004
- [15] J. S. Duncan, F. Lee, A. W. M. Smeulders and B. L. Zaret, A bending energy model for measurement of cardiac shape deformity, *IEEE Trans. Med. Imaging* 10, 1991, 307-320
- [16] A. C. Jalba, M. H. F. Wilkinson, J. B. T. M. Roerdink, M. M. Bayer and S. Juggins, Automatic diatom identification using contour analysis by morphological curvature scale spaces, *Machine vision and applications*, 16, 2005, 217-228
- [17] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision vol. 1*, Addison-Wesley, New York, 1992
- [18] O. I. Frette, G. Virnovsky, T. Hildebrand-Habel, Modelling the stability of thin water films using SEM images, To be published
- [19] A. Pressley, *Elementary differential geometry*, Springer, London, 2001

- [20] M. P. do Carmo, Differential geometry of curves and surfaces, Prentice-Hall, Englewood Cliffs, 1976
- [21] Tikhonov, A. N. and Arsenin, V . Y. Solutions of ill-posed problems, Scripta series in mathematics, Halsted Press, New York, 1977
- [22] P. J. Flynn, A. K. Jain, On reliable curvature estimation, Proceedings from CVPR '89., IEEE Computer Society Conference on Computer vision and pattern recognition, 1989
- [23] D. Coeurjolly, S. Miguet, L. Tougne, Discrete curvature based on osculating circle estimation, in Visual Form 2001, Proceedings from 4th international workshop on visual form, eds. C. Arcelli, L. P. Cordella, G. S. di Baja, Lecture notes in computer science, Springer, Berlin, 2001
- [24] M. Worring and A. W. M. Smeulders, Digital curvature estimation, CVGIP: Image understanding 58, 1993, 366-382
- [25] R. C. Gonzalez and R. E. Woods, Digital image processing 2nd ed., Prentice Hall, Upper Saddle River, 2002
- [26] A. Rosenfeld and A. C. Kak, Digital Picture Processing Vol 2, 2nd edition, Academic Press, New York, 1982
- [27] G. Wagenknecht, A contour tracing and coding algorithm for generating 2D contour codes from 3D classified objects, Pattern recognition, 40, 2007, 1294-1306
- [28] M. Sonka, V. Hlavac, R. Boyle, Image processing, analysis and machine vision 2nd ed., PWS Publishing, Pacific Grove, 1999

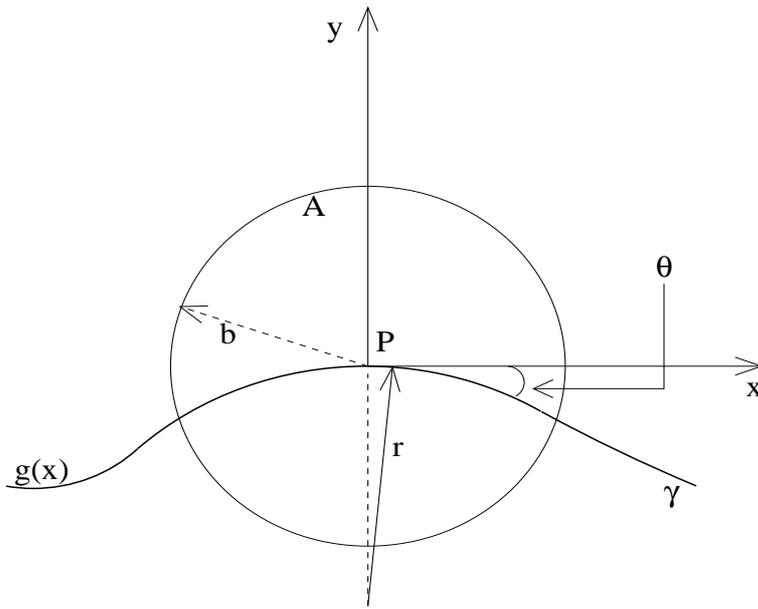


Fig. 1. There exists a linear relationship between the curvature of  $\gamma$  in  $P$  and the area of the circle above  $\gamma$ . See text for details.

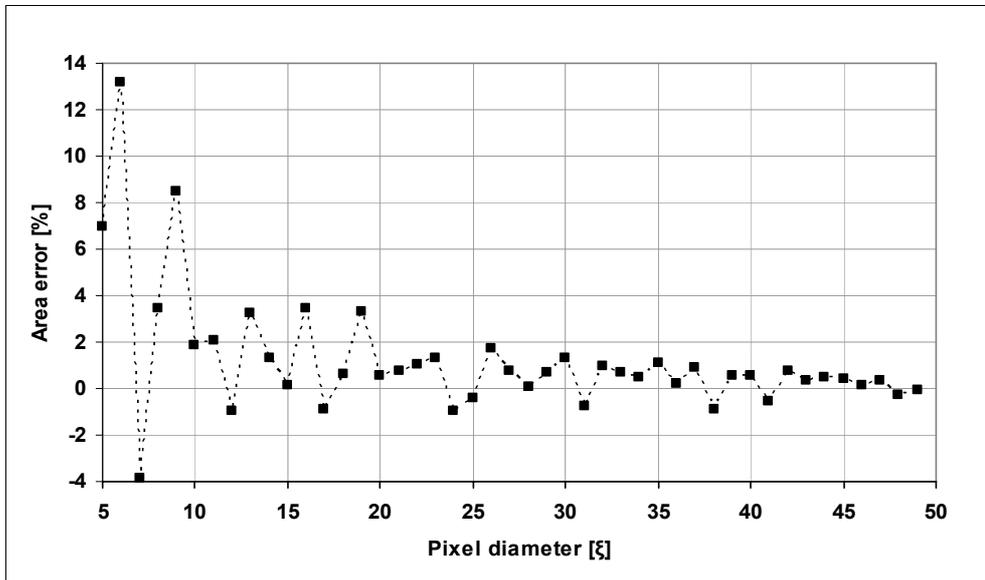


Fig. 2. Relative error of discrete circles generated from square pixels.  $\xi$  equals one pixel unit.

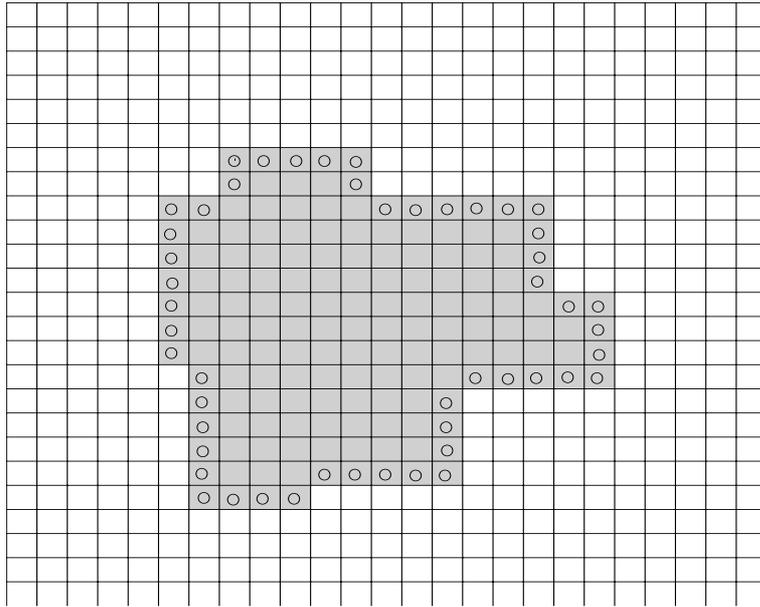


Fig. 3. Pixel based boundary. Points marked with O are boundary pixels.

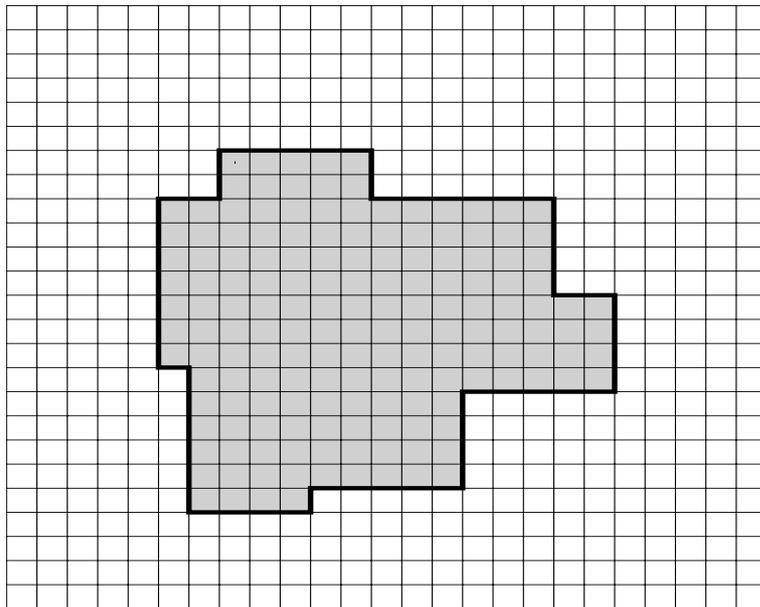


Fig. 4. Edge based boundary. The thick black line around the region represent the edges. Each boundary edge is shared between a region pixel and a background pixel. The thickness of the line is exaggerated.

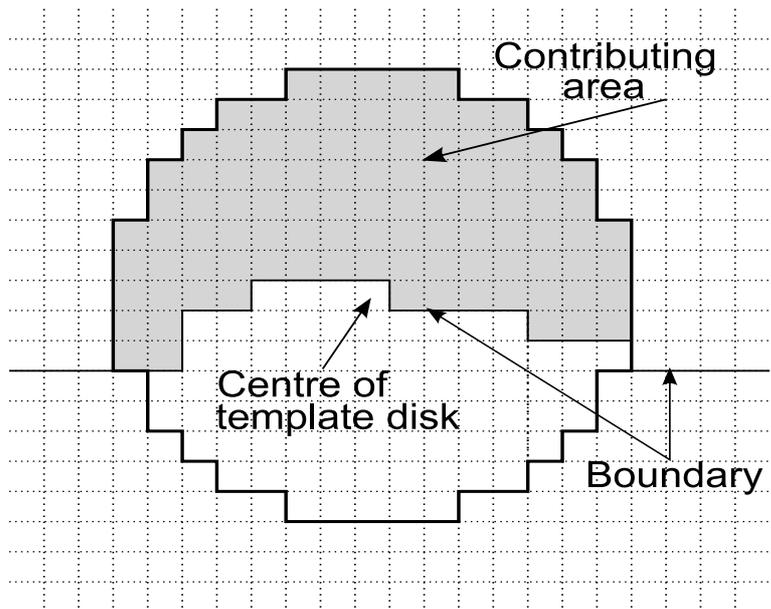


Fig. 5. Application of discrete template disk. In this example the pixel diameter is 15.

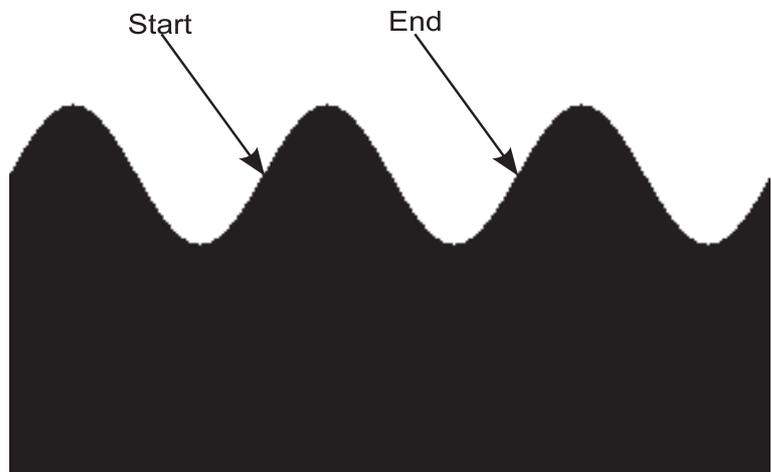


Fig. 6. Sinusoidal test image. Size 300x60 pixels. Curvature computed for the 100 pixels in the middle of the image from Start to End. Image is wider than test region to avoid boundary effects. Region is black, and background is white.

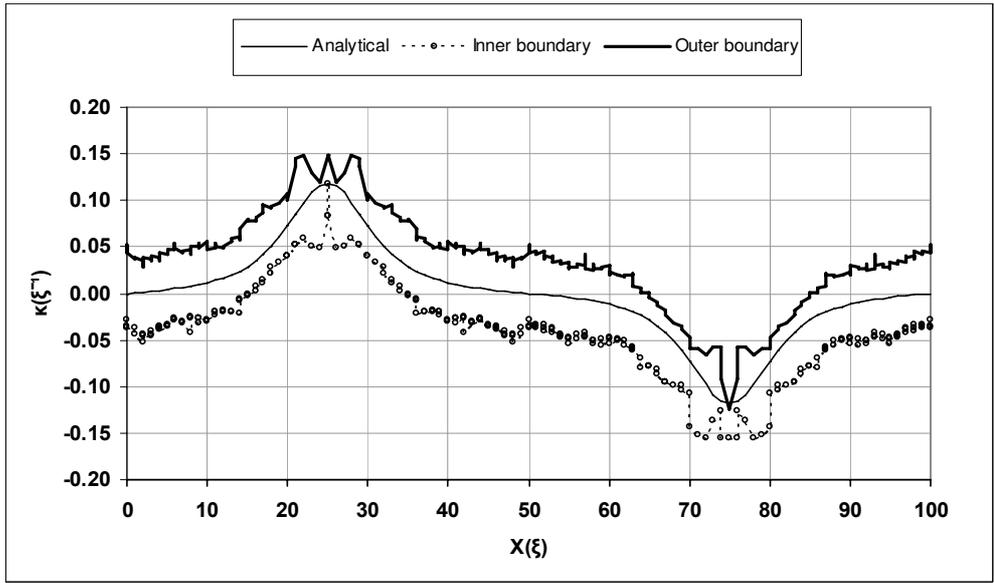


Fig. 7. Curvature bias due to boundary effects, with pixel diameter of 15 pixel. The two curves represent computed values using both inner and outer boundaries.

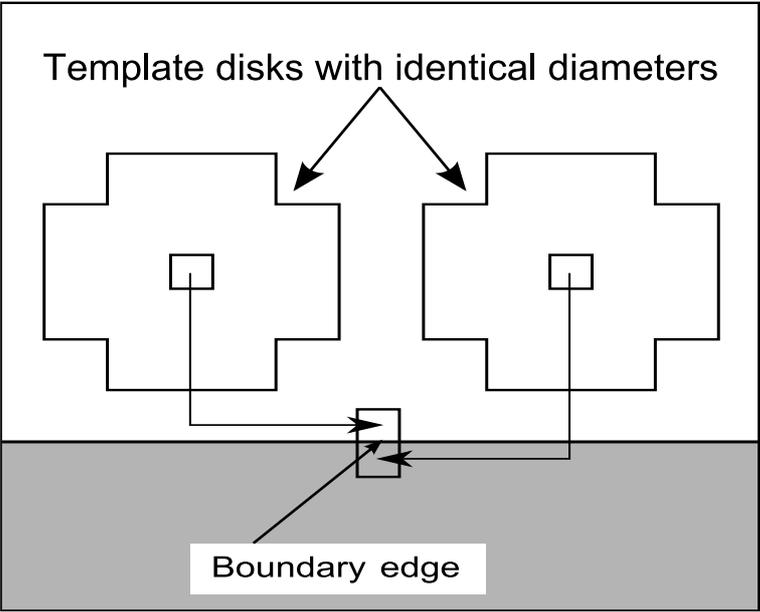


Fig. 8. To compute the curvature for a given edge the template disk is applied twice as shown in this schematic figure.

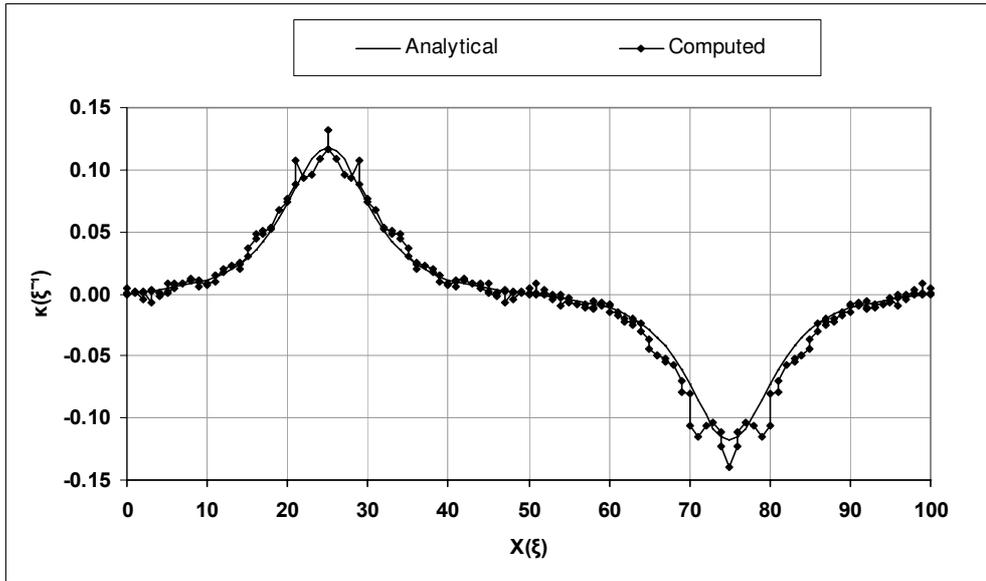


Fig. 9. Computed curvature values based on equation (14). Pixel diameter used for this simulation is 15.

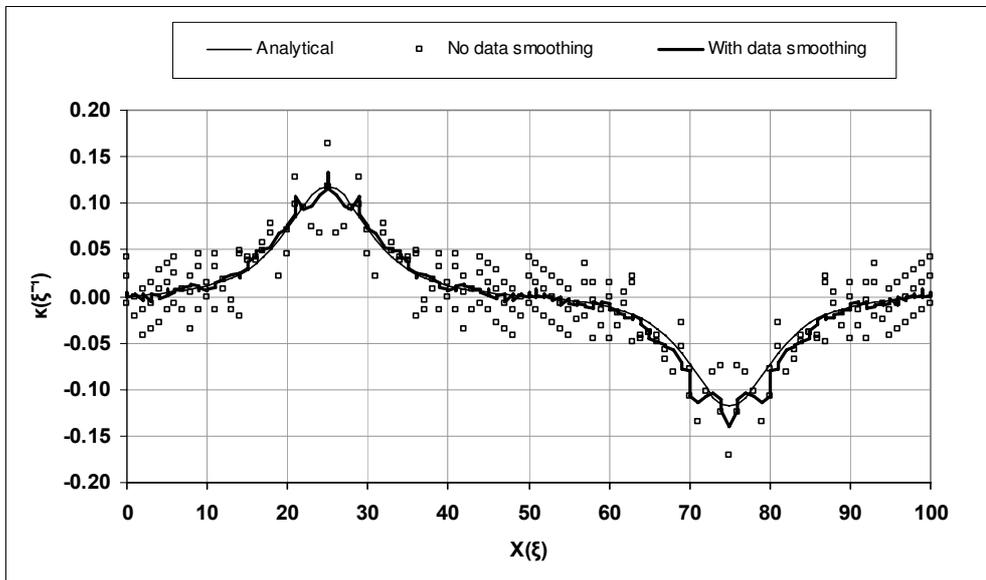


Fig. 10. Computed curvature values with and without data smoothing (noise removal). Template disk diameter used is 15 pixels.

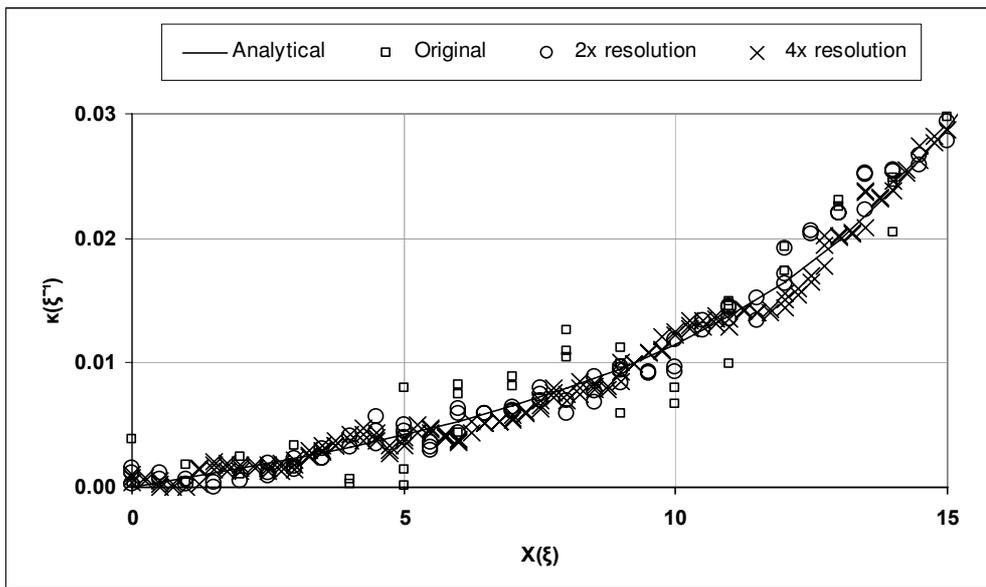


Fig. 11. Comparison of different resolutions. See text and table 1 for details.

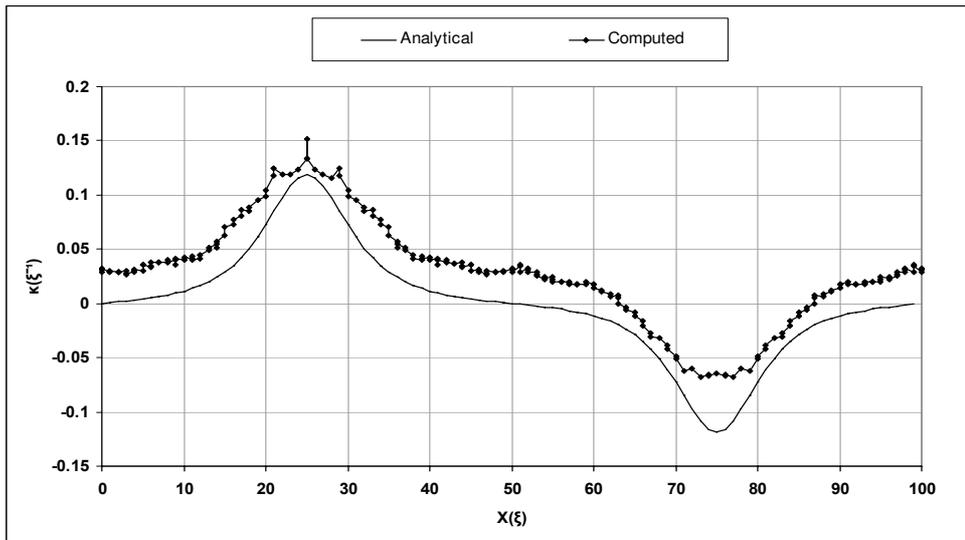


Fig. 12. Results using equation (17) with a pixel diameter of 19. A biased error is observed indicating that equation (17) is only suitable for pixel diameters of 15. Data smoothing has been applied using equation (16).

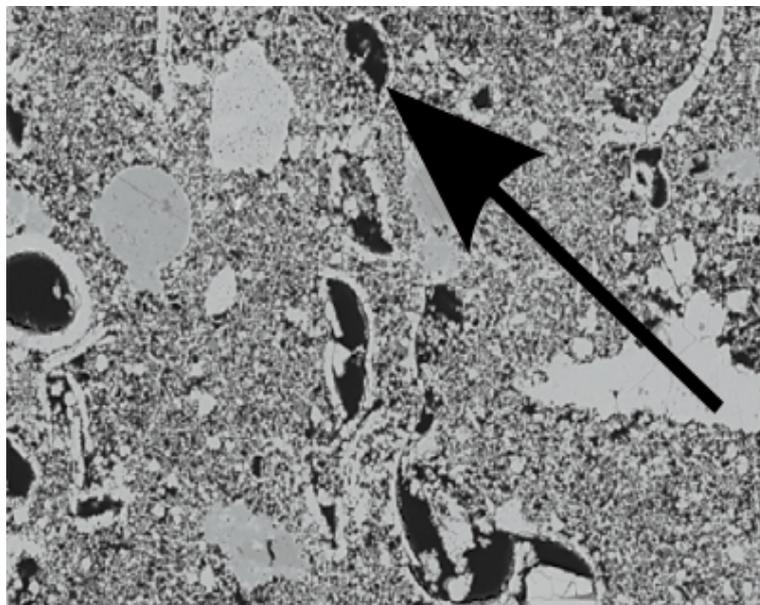


Fig. 13. Digital image of sedimentary rock (chalk). Resolution in image is  $0.313 \mu\text{m}/\text{pixel}$ . Arrow points at pore (figure 14) used in curvature computation.

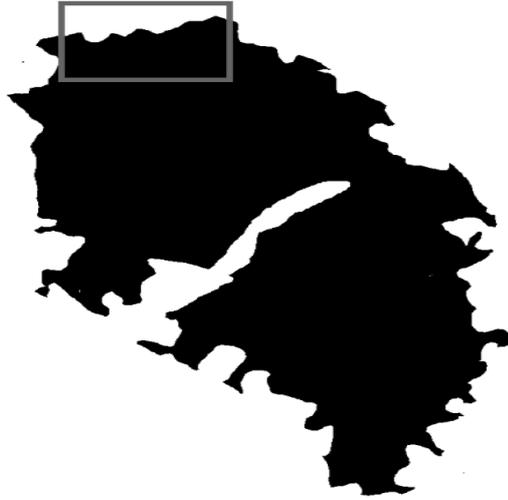


Fig. 14. Single pore (from figure 13) after image preprocessing. Boundary enclosed by grey rectangle used in curvature computation.

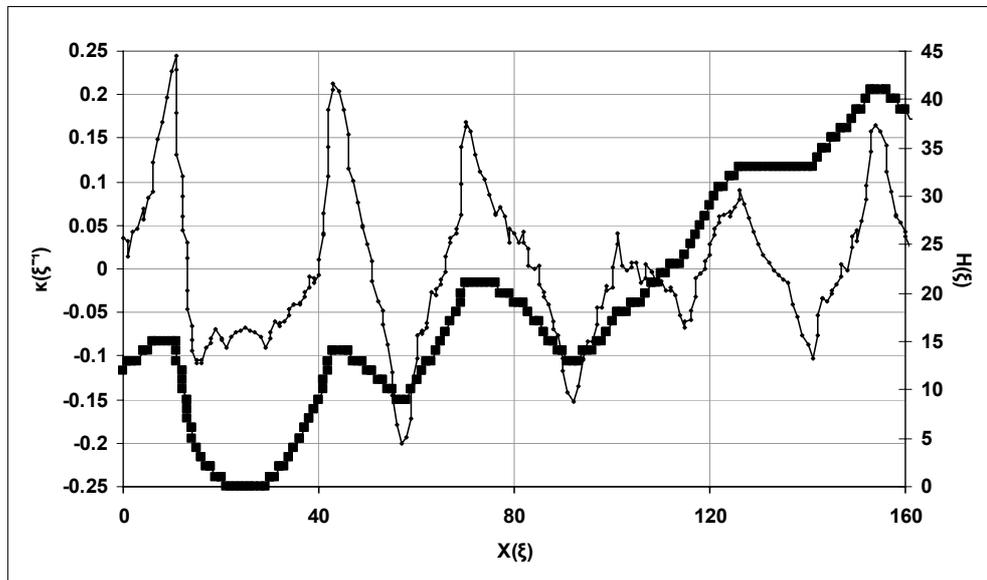


Fig. 15. Computed curvature [ $\text{pixel}^{-1}$ ] for subpart of pore (thin line). Boundary pixels of pore (thick line). Pixel diameter for template disk is 15. See text for further details.

<i>Image size [pixel]</i>	<i>Template disk diameter [pixel]</i>
100x60	15
200x120	31
400x240	61

Table 1

Data for resolution test.