Efficient Texture Analysis of Binary Images

Joachim Ohser[1], Bernd Steinbach[2], and Christian Lang[2]

May 25, 1998

Abstract

A new method of determining some characteristics of binary images is proposed based on a special linear filtering. This technique enables the estimation of the area fraction, the specific line length, and the specific integral of curvature. Furthermore, the specific length of the total projection is obtained, which gives detailed information about the texture of the image. The influence of lateral and directional resolution depending on the size of the applied filter mask is discussed in detail. The technique includes a method of increasing directional resolution for texture analysis while keeping lateral resolution as high as possible.

1 Introduction

A very powerful technique of image processing is the filtering of digital images. For example, high-pass or low-pass filtering can be used to eliminate noise at low or high frequencies, respectively. In the following we will show that linear filtering can also be applied in image analysis. The key idea of this paper is to formulate an algorithm for the computation of the global characteristics of binary structures which includes linear filtering as the basic step.

In this consideration a binary structure can be modelled by a planar random set which is assumed to be homogeneous, i.e. its distribution is invariant under translation of the set. In other words, the distribution properties of the structure are independent of the position of the observer. Particularly, the expectation of the area of a structure component α observed in a window W depends only on the size of W but not on its position. Such planar structures can be obtained from a planar section through a materials structure which consists of two phases. In this case α can be considered as the planar section of one of these phases.

The simplest global characteristic of the component α is the area fraction A_A; this is the mean area of α per unit area of the current section plane. Further global characteristics are the specific line length L_A and the specific integral of curvature C_A. The specific line length

---

[1] ITWM Kaiserslautern, Postfach 3049, D-67653 Kaiserslautern

[2] Freiberg University of Mining and Technology, D-09596 Freiberg

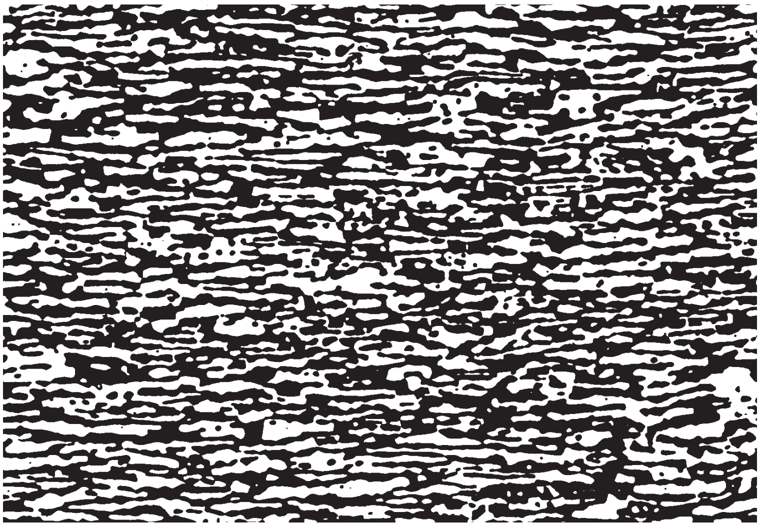

Figure 1: Binary image of a textured materials structure, micro structure of a rolled stainless steel, white phase – ferrite, black – austenite.

can be expressed in terms of the specific length of the total projection $p_A(\vartheta)$ of α with respect to the direction ϑ,

$$L_A = \frac{1}{2} \int_0^{2\pi} p_A(\vartheta)\,\mathrm{d}\vartheta, \tag{1}$$

see e.g. Ohser (1981). Notice that $p_A(\vartheta) = p_A(\vartheta + \pi)$. In image analysis this formula is well-known; it is the base for the determination of the specific line length. On the other hand, the specific total projection has its own meaning. The function $p_A(\vartheta)$ describes texture properties of α; it is closely related to the distribution of the direction of boundary of α. Let $R(\vartheta)$ denote the distribution function of the normal direction of a randomly chosen element of the boundary of α; $R(\vartheta)$ is called rose of directions. Then the equation

$$p_A(\vartheta) = \frac{L_A}{2} \int_0^{2\pi} |\cos(\vartheta - \varphi)|\,\mathrm{d}R(\varphi), \quad 0 \le \vartheta < 2\pi \tag{2}$$

relates these quantities to each other. This means that $p_A(\vartheta)$ and $R(\vartheta)$ can be understood as two representations of the same function. If $R(\vartheta) = R(\vartheta + \pi)$ then the specific total projection and the directional distribution function provide identical information about the texture of α. In this paper we confine ourselves to the texture characterisation of planar sections through spatial structures. For a detailed description of spatial texture analysis see Pohlmann *et al.* (1981). Adams & Field (1992), Adams (1993), and Beneš *et al.* (1994).

The specific integral of curvature C_A is closely related to the specific Euler number χ_A, $C_A = 2\pi\,\chi_A$, and if the component α consists of convex particles then C_A is up to a constant the specific particle number, $C_A = 2\pi\,N_A$.

From the literature about integral geometry it is well-known that the global characteristics

A_A, L_A, and C_A are the densities of the three random Minkowski measures of α. The properties of these quantities allow their estimation applying local methods such as linear filtering. The algorithm presented in this paper consists of three basic steps:

1. Linear filtering of the binary image which results in a gray-tone image,
2. generating the gray-tone histogram (integration step), and
3. computing the global characteristics from the data of the gray-tone histogram (analysis).

This estimation procedure is based on ideas of Serra (1982).

Note that the original image in this paper is binary which is transformed into a gray-tone image by filtering. Depending on the range of the corresponding filter mask the sizes of the gray-tone image and the gray-tone histogram can become very large. Thus in the implementation of the algorithm their explicit representation should be avoided. However, for a detailed description of the algorithm we temporarily assume that the grey-tone image as well as the grey-tone histogram are generated.

2 Linear Filtering of Binary Images

Consider a rectangular grid of points x_{ij}, $i = 0, \ldots, m$, $j = 0, \ldots, n$, with the digital resolution Δ and the aspect factor ζ of the grid, $\zeta > 0$; $\zeta \cdot \Delta$ is the distance of grid points in the horizontal direction and Δ is the distance in the vertical direction, $x_{ij} = (j\zeta\Delta, i\Delta)$. A discretisation of the α-phase observed in the rectangular sampling window $W = [0, m\Delta] \times [0, n\zeta\Delta]$ is the binary image given by the matrix $B = (b_{ij})$ of pixels

$$b_{ij} = 1_\alpha(x_{ij}), \quad i = 0, \ldots, m,\ j = 0, \ldots, n, \tag{3}$$

where $1_\alpha(x)$ denotes the indicator function. Let $F_\nu = (f_{k\ell})$ be the mask of a linear filter of finite size ν, $\nu = 0, 1, 2, \ldots$ The coefficients of F_ν are given by $f_{k\ell} = 2^{k+(\nu+1)\ell}$, $k = 0, \ldots, \nu$, $\ell = 0, \ldots, \nu$. We obtain

$$F_0 = (1), \quad F_1 = \begin{pmatrix} 1 & 4 \\ 2 & 8 \end{pmatrix}, \quad F_2 = \begin{pmatrix} 1 & 8 & 64 \\ 2 & 16 & 128 \\ 4 & 32 & 256 \end{pmatrix}, \ldots$$

Linear filtering of the binary image B by F_ν is nothing other than the convolution of B and F_ν, $G = B * F_\nu$. The filter response is a digital grey-tone image $G = (g_{ij})$,

$$g_{ij} = \sum_{k=0}^{\nu} \sum_{\ell=0}^{\nu} b_{i+k,j+\ell} f_{k\ell}$$

for $i = 0, \ldots, m - \nu$, $j = 0, \ldots, n - \nu$. Notice that $(\nu + 1)^2$ is the number of bits per pixel. Furthermore, we remark that the components g_{ij} of G cannot be computed for $i > m - \nu$

and $j > n - \nu$, i.e. the filter response G is free of edge effects within the partial window $W_\nu = [0, (m-\nu)\Delta] \times [0, (n-\nu)\zeta\Delta]$.

The linear filter mentioned above is not original since this kind of encoding is used quite often in mathematical morphology. Its use in image processing was first suggested by the Ecole des Mines de Paris. However, the proposed application of this filter in stereological estimation is original.

3 Generating the Grey-Tone Histogram

The next step in the computation of the global characteristics is the determination of the grey-tone histogram h of the image G. The component h_k of h is the number of pixels of G for which the grey-tone is k. This means the k indicates a pixel configuration of the corresponding binary image B, and h_k is the number of this configuration. For example, if $\nu = 1$ then $k = 7$ is assigned to the point configuration

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}.$$

Consequently, h_7 is the number N of these configurations within the binary image B,

$$h_7 = N \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}.$$

We will use the function $\delta(g,k)$ to indicate the integers g which are equal to the index k; δ is Kronecker's Delta, $\delta(g,k) = 1$ for $g = k$, and $\delta(g,k) = 0$ otherwise. Then the h_k can be expressed as

$$h_k = \sum_{i=0}^{m-\nu} \sum_{j=0}^{n-\nu} \delta(g_{ij}, k)$$

for $k = 0, \ldots, \mu - 1$. Notice that the sum over the h_k is the total pixel number within the partial window W_ν, $\sum h_k = (m - \nu + 1)(n - \nu + 1)$. Thus

$$A(W_\nu) = \zeta\, \Delta^2 \sum_{k=0}^{\mu-1} h_k,$$

where $\mu = 2^{(\nu+1)(\nu+1)}$. The grey-tone histogram h of the filtered image G comprises the complete information about the global characteristics available from the image. Thus h is the base of the estimation of A_A, L_A, and C_A.

4 Analysis

4.1 Area, Area Fraction

The area of α within the window W can be determined from the number of pixels x_{ij} for which the value of g_{ij} is odd,

$$A(\alpha \cap W_\nu) = \zeta\,\Delta^2 \sum_{k=0}^{\mu/2-1} h_{2k+1}. \tag{4}$$

Therefore,

$$\widehat{A_A} = \frac{\sum_{k=0}^{\mu/2-1} h_{2k+1}}{\sum_{k=0}^{\mu-1} h_k} \tag{5}$$

is an unbiased estimator of the area fraction. Let $k|\kappa$ denote the bitwise *or* of the integers k and κ, e.g. $19|26 = 10011|11010 = 11011 = 27$. With the help of this notation Eq. (**??**) can be rewritten as

$$\widehat{A_A} = \frac{\zeta\,\Delta^2}{A(W_\nu)} \sum_{k=0}^{\mu-1} h_k\,\delta(k, k|1). \tag{6}$$

Of course, this seems to be a quite complicated estimator for such a simple quantity as the area fraction. However, the description of this approach is very instructive in understanding the method of statistical estimation for the other global characteristics as well.

4.2 Length of Total Projection, Specific Line Length

The specific total projection $p_A(\vartheta)$ can be estimated from the probability that a point x lies in the component α while the point $y = x + (r, \vartheta)$ lies in the complement of α. Here r denotes the distance $\|y - x\|$ of the points x and y. For a fixed direction ϑ this probability is a function of the distance r and for $r \to 0$ the slope of this function converges to $p_A(\vartheta)$. This leads one to construct an estimator for $p_A(\vartheta)$ in the following way: Let ϑ_ℓ denote the direction of a pair of grid points of the filter mask F_ν; r_ℓ is the grid distance corresponding to this direction, $\ell = 0, \ldots, 8\nu - 1$. The specific length of the total projection $p_A(\vartheta)$ of the α-phase with respect to the direction ϑ_ℓ can be computed from the grey-tone histogram h using

$$\widehat{p_A}(\vartheta_\ell) = \frac{\zeta\Delta^2}{r_\ell\,A(W_\nu)} \sum_{k=0}^{\mu-1} h_k\,\delta(k, k|\kappa_{0,\ell})\,[1 - \delta(k, k|\kappa_{1,\ell})], \quad \ell = 0, \ldots, 8\nu - 1, \tag{7}$$

where the masks $\kappa_{0,\ell}$ and $\kappa_{1,\ell}$ indicate the pair of grid points of the filter mask F_ν concerning ϑ_ℓ. The directions ϑ_ℓ, the distances r_ℓ, and the masks $\kappa_{0,\ell}$ and $\kappa_{1,\ell}$ are given in Table **??** for the special case $\nu = 1$. It has been shown by Serra (1982) that this estimator is asymptotically unbiased for $\Delta \to 0$. The estimation $\widehat{p_A}(\vartheta)$ of the specific total projection is an alternative to the much more complicated measurement of the rose of directions suggested e.g. by Chaix & Grillon (1996).

As a discrete version of Eq. (??) we obtain an estimator of the specific line length L_A,

$$\widehat{L_A} = \frac{1}{2} \sum_{\ell=0}^{8\nu-1} (\vartheta_{\ell+1} - \vartheta_\ell)\, \widehat{p_A}(\vartheta_\ell). \tag{8}$$

This estimator is always biased. The bias depends on the lateral resolution as well as the directional discretisation. The lateral resolution is given by the $1/r_i$ where $r_i \approx \nu\Delta$, and the directional discretisation is represented by the differences $\vartheta_{\ell+1} - \vartheta_\ell \approx \pi/(4\nu)$. Thus for a small size ν of the filter mask the lateral resolution is high, and the lateral resolution decreases for increasing ν. On the other hand, the fineness of directional discretisation (directional resolution) increases with increasing ν. In other words, by the parameter ν one can choose between high digital resolution or high directional resolution. Therefore, the problem consists in finding an appropriate ν depending on the given structure α and the required accuracy of estimation.

ℓ	ϑ_ℓ	r_ℓ	$\kappa_{0,\ell}$	$\kappa_{1,\ell}$
0	0	$\Delta\zeta$	1	4
1	$\operatorname{arccot}\zeta$	$\Delta\sqrt{1+\zeta^2}$	1	8
2	$\pi/2$	Δ	4	8
3	$\pi - \operatorname{arccot}\zeta$	$\Delta\sqrt{1+\zeta^2}$	4	2
4	π	$\Delta\zeta$	8	2
5	$\pi + \operatorname{arccot}\zeta$	$\Delta\sqrt{1+\zeta^2}$	8	1
6	$3\pi/2$	Δ	2	1
7	$2\pi - \operatorname{arccot}\zeta$	$\Delta\sqrt{1+\zeta^2}$	2	4

Table 1: The parameters ϑ_ℓ, r_ℓ, $\kappa_{0,\ell}$, and $\kappa_{1,\ell}$ for a filter mask of size $\nu = 1$.

It should be noted that the same estimators can be computed by reverting the image, so that the upper-right corner becomes the lower-left corner. Then one can use for each estimation the mean of the proposed estimators computed on the two images, in order to use all the information in the original image.

4.3 Euler Number, Integral of Curvature per Unit Area

In this section we describe how to count objects in a planar image where the number of objects is indicated by the Euler number χ. Discrete versions of the Euler number χ of α in the planar sampling window W are closely related to planar graphs associated with the considered point grid. Starting from a point grid, graphs can simply be constructed by adding edges. Several modes of construction lead to various types of graphs. Examples are given in Figure **??**; further ones can be easily imagined, see e.g. Serra (1982, p. 174) and Heijmans (1994, pp. 326–328). A discrete version of the Euler number of α in W is the Euler number of the corresponding partial graph which can be obtained from one of Euler's famous relations

$$\chi_0 = v - e + f, \tag{9}$$

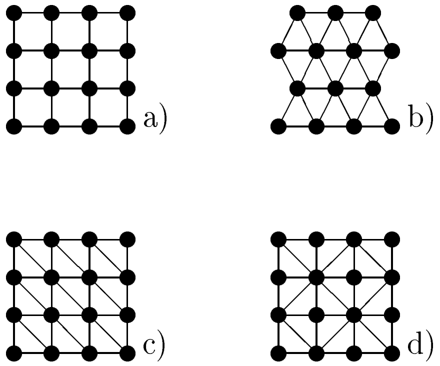

Figure 2: Typical graphs generated for rectangular point grids. a) The simple rectangular graph which is also called 4-connected graph, b) hexagonal-I (6-connected), c) hexagonal-II (6-connected), d) octagonal-I (4-8-connected).

where v, e, and f denote the number of vertices (points), the number of edges, and the number of elementary faces (squares or triangles), respectively. We will use χ_0 to denote the Euler number of the graph of the α-phase restricted to the given point grid.

For rectangular and 8-connected graphs the Euler number of α is in general not the negative Euler number of the complement of α, $\chi_0(\alpha) \neq -\chi_0(\alpha^c)$. In the case of graph d) the Euler number χ_0 depends on translations of the graph, see Figure **??**. Note that from a topological point of view the hexagonal graphs b) and c) are identical (up to edge effects). They do not have the above disadvantages; $\chi_0(\alpha) = -\chi_0(\alpha^c)$ is valid, and χ_0 remains independent of translations of the graph. For this reason, we choose a hexagonal graph as an appropriate base for determining the Euler number, see Serra (1982, pp. 186, 201).

In the following we confine ourselves to the case of hexagonal-II grid. For $\nu = 1$ we can write

$$\begin{aligned} v &= N\begin{pmatrix} 1 & \cdot \\ \cdot & \cdot \end{pmatrix}, \\ e &= N\begin{pmatrix} 1 & 1 \\ \cdot & \cdot \end{pmatrix} + N\begin{pmatrix} 1 & \cdot \\ 1 & \cdot \end{pmatrix} + N\begin{pmatrix} 1 & \cdot \\ \cdot & 1 \end{pmatrix}, \\ f &= N\begin{pmatrix} 1 & \cdot \\ 1 & 1 \end{pmatrix} + N\begin{pmatrix} 1 & 1 \\ \cdot & 1 \end{pmatrix}. \end{aligned}$$

Here $N()$ denotes the number of point configurations in the binary image where "$\cdot$" indicates that the binary value of the corresponding pixel is either 0 or 1. For example, v is the number of configurations for which the upper left grid point is 1 and the others are arbitrary, i.e. v is simply the number of grid points in W_1 for which $b_{ij} = 1$.

The above equations can be rewritten as

$$v = (h_1 + h_3 + h_5 + h_7 + h_9 + h_{11} + h_{13} + h_{15}),$$

$$\begin{aligned} e &= (h_5 + h_7 + h_{13} + h_{15}) + (h_3 + h_7 + h_{11} + h_{15}) + (h_9 + h_{11} + h_{13} + h_{15}), \\ f &= (h_{11} + h_{15}) + (h_{13} + h_{15}), \end{aligned}$$

and the Euler relation (**??**) implies

$$\chi_0 = h_1 - h_7. \tag{10}$$

This discrete version of the Euler number can be assigned the direction $\vartheta_0 = 0$. For the directions $\vartheta_2 = \pi/2$, $\vartheta_4 = \pi$, and $\vartheta_6 = 3\pi/2$ we have

$$\chi_2 = h_2 - h_{11}, \quad \chi_4 = h_8 - h_{14}, \quad \chi_6 = h_4 - h_{13}. \tag{11}$$

From the average of the χ_ℓ one obtains an estimation of the integral of mean curvature per unit area,

$$\begin{aligned} \widehat{C_A} &= \frac{\pi}{2\,A(W_\nu)} \sum_{i=0}^{3} \chi_{2\ell} \\ &= \frac{\pi}{2\,A(W_\nu)} (h_1 + h_2 + h_4 + h_8 - h_7 - h_{11} - h_{13} - h_{14}), \end{aligned} \tag{12}$$

which is asymptotically unbiased for $\Delta \to 0$, see Serra (1982). This estimator can simply be generalized for each $\nu > 0$ when replacing the indices 1, 2, 4, 8 in (**??**) by $1,\ 2^\nu,\ 2^{\nu(\nu+1)},\ 2^{\nu(\nu+2)}$, and the indices 7, 11, 13, 14 have to be replaced by $1 + 2^\nu + 2^{\nu(\nu+1)},\ 1 + 2^\nu + 2^{\nu(\nu+2)},\ 1 + 2^{\nu(\nu+1)} + 2^{\nu(\nu+2)},\ 2^\nu + 2^{\nu(\nu+1)} + 2^{\nu(\nu+2)}$, respectively.

5 Example

As an example we consider a binary image B consisting of 16×16 pixels,

$$B = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}.$$

The corresponding hexagonal-II graph of B is shown in Figure **??**. The convolution of B

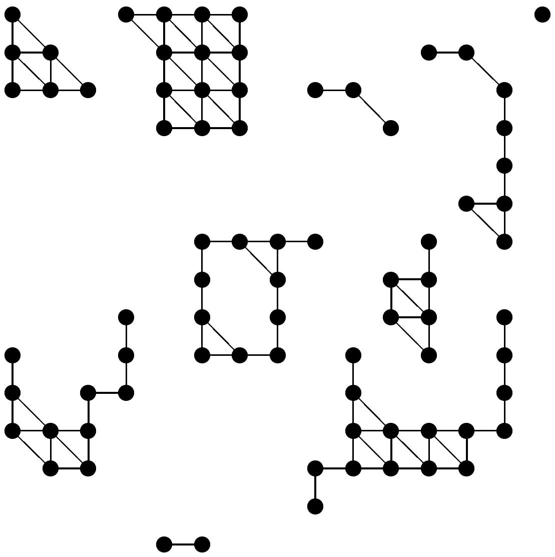

Figure 3: Hexagonal-II graph for the example image B.

with the filter mask F_1 yields a grey-tone image G with four bits per pixel. For the matrix G and the grey-tone histogram h of G we obtain

$$G = B * F_1 = \begin{pmatrix}
8 & 2 & 0 & 8 & a & a & a & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & \cdot \\
c & b & 2 & 4 & d & f & f & 3 & 0 & 0 & 0 & 8 & a & 2 & 4 & \cdot \\
c & f & b & 2 & c & f & f & 3 & 8 & a & 2 & 4 & 5 & 9 & 2 & \cdot \\
4 & 5 & 5 & 1 & c & f & f & 3 & 4 & 5 & 9 & 2 & 0 & c & 3 & \cdot \\
0 & 0 & 0 & 0 & 4 & 5 & 5 & 1 & 0 & 0 & 4 & 1 & 0 & c & 3 & \cdot \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & e & 3 & \cdot \\
0 & 0 & 0 & 0 & 0 & 8 & a & a & a & 2 & 0 & 8 & 6 & d & 3 & \cdot \\
0 & 0 & 0 & 0 & 0 & c & 7 & d & 7 & 1 & 8 & e & 3 & 4 & 1 & \cdot \\
0 & 0 & 0 & 8 & 2 & c & 3 & c & 3 & 0 & c & f & 3 & 8 & 2 & \cdot \\
8 & 2 & 0 & c & 3 & c & b & e & 3 & 8 & 6 & d & 3 & c & 3 & \cdot \\
c & 3 & 8 & e & 3 & 4 & 5 & 5 & 1 & c & 3 & 4 & 1 & c & 3 & \cdot \\
c & b & e & 7 & 1 & 0 & 0 & 0 & 0 & c & b & a & a & e & 3 & \cdot \\
4 & d & f & 3 & 0 & 0 & 0 & 0 & 8 & e & f & f & f & 7 & 1 & \cdot \\
0 & 4 & 5 & 1 & 0 & 0 & 0 & 0 & c & 7 & 5 & 5 & 5 & 1 & 0 & \cdot \\
0 & 0 & 0 & 0 & 8 & a & 2 & 0 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{pmatrix}, \quad h = \begin{pmatrix} 70 \\ 12 \\ 13 \\ 21 \\ 13 \\ 12 \\ 2 \\ 5 \\ 16 \\ 2 \\ 11 \\ 5 \\ 19 \\ 5 \\ 7 \\ 12 \end{pmatrix}$$

where the grey-values of G are presented in a hexadecimal code. Note that the right column and the lower row of G cannot be determined free of edge-effects. From the components of

h we have

$$\begin{array}{lll}
A(W_1) = 225\,\zeta\Delta^2, & \widehat{p_A}(\vartheta_0) = 40\,\frac{\Delta}{A(W_\nu)} & \chi_0 = 7, \\
A(\alpha \cap W_1) = 74\,\zeta\Delta^2, & \widehat{p_A}(\vartheta_1) = 50\,\frac{\zeta\Delta}{\sqrt{1+\zeta^2}\,A(W_\nu)}, & \\
 & \widehat{p_A}(\vartheta_2) = 32\,\frac{\zeta\Delta}{A(W_\nu)}, & \chi_2 = 8, \\
 & \widehat{p_A}(\vartheta_3) = 49\,\frac{\zeta\Delta}{\sqrt{1+\zeta^2}\,A(W_\nu)}, & \\
 & \widehat{p_A}(\vartheta_4) = 42\,\frac{\Delta}{A(W_\nu)}, & \chi_4 = 9, \\
 & \widehat{p_A}(\vartheta_5) = 53\,\frac{\zeta\Delta}{\sqrt{1+\zeta^2}\,A(W_\nu)}, & \\
 & \widehat{p_A}(\vartheta_6) = 33\,\frac{\zeta\Delta}{A(W_\nu)}, & \chi_6 = 8, \\
 & \widehat{p_A}(\vartheta_7) = 50\,\frac{\zeta\Delta}{\sqrt{1+\zeta^2}\,A(W_\nu)}. &
\end{array}$$

We remark that the variation of the Euler number is a pure edge effect; the mean Euler number is 8. The binary image consists of 10 objects, two of them are intersected by the edge of the sampling window W and a further one is not simply connected. Furthermore, the differences between $\widehat{p_A}(\vartheta_\ell)$ and $\widehat{p_A}(\vartheta_{\ell+4})$, $\ell = 0, \ldots, 3$, are also caused by edge effects only.

If the basic grid is a square grid, $\zeta = 1$, then we obtain the estimations

$$\widehat{A_A} = 0.329, \qquad \widehat{L_A} = 0.506/\Delta, \qquad \widehat{C_A} = 0.223/\Delta^2.$$

We remark that stereological interpretation of these values yields estimates of the volume fraction, the specific surface area, and the specific mean curvature.

6 Digital Resolution and Fractal Dimension

It seems to be impossible to increase the lateral resolution as well as the directional resolution. In fact, improved estimation of the specific line length L_A and the texture representation $p_A(\vartheta)$ basing on high lateral and directional resolution is only possible if the influence of the lateral resolution is known. If the influence of lateral resolution is low then $\widehat{p_A}(r, \vartheta) \approx \widehat{p_A}(\Delta, \vartheta)$ where $\widehat{p_A}(\vartheta, r)$ and $\widehat{p_A}(\vartheta, \Delta)$ are estimates of $p_A(\vartheta)$ at (r, ϑ) and (Δ, ϑ), respectively. Thus

$$\log(r \cdot \widehat{p_A}(r, \vartheta)) \approx \log(\Delta \cdot \widehat{p_A}(\Delta, \vartheta)) + \log \frac{r}{\Delta}. \tag{13}$$

If the influence of the lateral resolution cannot be neglected then Eq. (**??**) must be modified. Simple extension yields the approach

$$\log(r \cdot \widehat{p_A}(r, \vartheta)) \approx \log(\Delta \cdot \widehat{p_A}(\Delta, \vartheta)) + (1 + d)\, \log \frac{r}{\Delta} \tag{14}$$

where d is an appropriately chosen constant, $d \leq 0$. In this linear model the function $\widehat{p_A}(\Delta, \vartheta)$ is independent of the lateral resolution $1/r$, and the slope d is assumed to be independent of the direction ϑ.

The unknown values of $\widehat{p_A}(\Delta, \vartheta)$ and d can be found as the least-squares solution to the overdetermined set of linear equations

$$\log \widehat{p_A}(r_\ell^\nu, \vartheta_\ell^\nu) = \log \widehat{p_A}(\Delta, \vartheta_\ell^\nu) + d \log \frac{r_\ell^\nu}{\Delta}, \quad \ell = 0, \ldots, 8\nu - 1, \quad \nu > 0 \tag{15}$$

(loglinear regression). Here, the r_ℓ^ν and the ϑ_ℓ^ν denote the distances and the directions, respectively, for chosen size ν. From the $\widehat{p_A}(\Delta, \vartheta)$ one can obtain an improved estimator of the specific line length L_A,

$$\widetilde{L_A} = \frac{1}{2} \sum_{\ell=0}^{8\nu-1} (\vartheta_{\ell+1} - \vartheta_\ell)\, \widetilde{p_A}(\vartheta_\ell) \tag{16}$$

with $\widetilde{p_A}(\vartheta_\ell) = \widehat{p_A}(\Delta, \vartheta_\ell)$. This estimator is robust with respect to variation of the size ν because it is related to the maximum lateral resolution $1/\Delta$. This means, in some bounds the directional resolution can be increased without a considerable reduction of the lateral resolution.

The slope d describes influence of the lateral resolution on the estimates $\widehat{L_A}$ and $\widehat{p_A}(\vartheta)$. Clearly, $\widehat{L_A} \to \widetilde{L_A}$ and $\widehat{p_A}(\vartheta) \to \widetilde{p_A}(\vartheta)$ as $d \to 0$. Furthermore, if the component α can be described by a fractal set then $d > 0$ and the real number $\widetilde{D} = 1 + d$ can be used as an estimator of the fractal dimension D of α; D is called correlation dimension, see Stoyan & Stoyan (1994, p. 61).

7 Algorithm and Implementation

7.1 The Size of the Histogram

The size of a histogram μ depends on the size ν of the filter mask, $\mu = 2^{(\nu+1)(\nu+1)}$. This size becomes very large just for small ν. Therefore, the computation of the histogram h as a substep of the algorithm is only practicable for $\nu \leq 2$. In this case, the size of h is always smaller than the size of the binary image. This means that the computation of h can involve a considerable reduction of the data size. Otherwise, if we enlarge the size of the filter mask, then the size of the histogram explodes, e.g. $\mu(\nu = 5) = 33,554,432$ or $\mu(\nu = 7) = 562,949,953,421,312$. Thus we need a modified algorithm which avoids the computation of both the grey-tone image and the histogram.

7.2 Area, Area Fraction

Of course, it is trivial to count the black pixels in the window W_ν of the binary image. The computation of the area fraction runs in $O((m - \nu)(n - \nu))$ time, which is linear in the size of the image.

7.3 Specific Length of Total Projection

The function `projection` calculates the vector of 8ν texture coefficients $\widehat{p_A}(\vartheta_\ell)$ for a given binary image B and returns a pointer to this vector.

```
double *projection(int ny, double zeta, int m, int n, bin_image B)
{
  double *p, *r;
  int i, j, l;
  int all = (m-ny)*(n-ny);
  p = (double *)malloc(8*ny*sizeof(double));
  r = (double *)malloc(4*ny*sizeof(double));
  for(l=0;l<8*ny;l++)  p[l] = 0.0;
  for(i=0;i<n-ny;i++)
    for(j=0;j<m-ny;j++)
    {
      if(B[i][j])        for(l=0;l<ny;l++) if(!B[i+l][j+ny])    p[l]++;
      if(B[i][j])        for(l=0;l<ny;l++) if(!B[i+ny][j+ny-l]) p[l+ny]++;
      if(B[i][j+ny])     for(l=0;l<ny;l++) if(!B[i+ny][j+ny-l]) p[l+2*ny]++;
      if(B[i][j+ny])     for(l=0;l<ny;l++) if(!B[i+ny-l][j])    p[l+3*ny]++;
      if(B[i+ny][j+ny]) for(l=0;l<ny;l++) if(!B[i+ny-l][j])    p[l+4*ny]++;
      if(B[i+ny][j+ny]) for(l=0;l<ny;l++) if(!B[i][j+l])       p[l+5*ny]++;
      if(B[i+ny][j])     for(l=0;l<ny;l++) if(!B[i][j+l])       p[l+6*ny]++;
      if(B[i+ny][j])     for(l=0;l<ny;l++) if(!B[i+l][j+ny])    p[l+7*ny]++;
    }
  for(l=0;l<ny;l++)                                    // radius part 1
    r[l] = sqrt(l*l+(ny*zeta)*(ny*zeta));
  for(l=0;l<=ny;l++)                                   // radius part 2
    r[l+ny] = sqrt(ny*ny+(ny-l)*zeta*(ny-l)*zeta);
  for(l=1;l<2*ny;l++) r[2*ny+l] = r[2*ny-l];          // sym. copy of radius
  for(l=0;l<8*ny;l++) p[l] = (p[l]*zeta)/(all*r[l%(4*ny)]);
  free(r);
  return p;
}
```

The run time of `projection` is dominated by the nested for-loops. There are at most $8\nu(m-\nu)(n-\nu)$ iterations. Each iteration takes a constant amount of time. Therefore, `projection` runs in $O(\nu(m-\nu)(n-\nu))$ time. For the final calculation of $\widehat{p_A}(\vartheta_\ell)$ we need 8ν coefficients r_ℓ. The formulas for these coefficients can be picked up in the program at `// radius part 1` for the interval $0 \le \ell \le \nu$ and at `// radius part 2` for the interval $\nu \le \ell \le 2\nu$. Additional, we use the symmetry properties

$$r_{2\nu+\ell} = r_{2\nu-\ell}, \qquad 1 \le \ell \le 2\nu - 1 \tag{17}$$

and

$$r_{4\nu+\ell} = r_\ell, \qquad 0 \le \ell \le 4\nu - 1 \tag{18}$$

for the distances of the corresponding pairs of points.

7.4 Specific Boundary Length

The function `length` returns the specific line length $\widehat{L_A}$ using the vector of 8ν texture coefficients $\widehat{p_A}(\vartheta_\ell)$ calculated by the previous function `projection`.

```
double length(int ny, double zeta, double *p)
{
  int l;
  double l_a = 0, *delta_theta;
  delta_theta = (double *)malloc(4*ny*sizeof(double));
  for(l=0;l<ny;l++)                                 // theta part 1
    delta_theta[l] = atan2(l+1, zeta*ny);
  for(l=0;l<ny;l++)                                 // theta part 2
    delta_theta[ny+l] = atan2(ny, zeta*(ny-(l+1)));
  for(l=2*ny-1;l>0;l--)                             // delta_theta
    delta_theta[l] = delta_theta[l]-delta_theta[l-1]
  for(l=0;l<2*ny;l++)                               // sym. copy of delta_theta
    delta_theta[2*ny+l] = delta_theta[2*ny-(l+1)]
  for(l=0;l<8*ny;l++)                               // length
    l_a += p[l]*delta_theta[l%(4*ny)];
  free(delta_theta);
  return l_a/2;
}
```

The loops of `length` are iterated at most 8ν times. Each iteration takes a constant amount of time. Therefore, `length` runs in $O(\nu)$ time. In the function `length` you find the formulas to calculate ϑ_ℓ for the first two intervals and $\Delta\vartheta_\ell = \vartheta_{\ell+1} - \vartheta_\ell, 0 \le \ell \le 2\nu - 1$. As for the distances r_ℓ, there is also a symmetry property for $\Delta\vartheta_\ell$.

$$\Delta\vartheta_{2\nu+\ell} = \Delta\vartheta_{2\nu-1-\ell}, \qquad 0 \le \ell \le 2\nu - 1. \tag{19}$$

Notice that in comparison with (**??**) the symmetry line is shifted by a half index and the symmetry interval is enlarged by one. However, the coefficients $\Delta\vartheta_\ell$ are also periodically as the r_ℓ,

$$\Delta\vartheta_{4\nu+l} = \Delta\vartheta_\ell, \qquad 0 \le \ell \le 4\nu - 1, \tag{20}$$

see also Eq. (**??**), so we calculate only half of the coefficients.

7.5 Calculation of the Specific Integral of Curvature

Finally the function `curvature` calculates and returns the specific integral of curvature per unit area $\widehat{C_A}$ for a given binary image B and a selected dimension of filter ν.

```
double curvature(int ny, int m, int n, bin_image B)
{
  int count, i, j, h = 0;
  for(i=0;i<n-ny;i++)
    for(j=0;j<m-ny;j++)
    {
      count = B[i][j]+B[i+ny][j]+B[i][j+ny]+B[i+ny][j+ny];
      if(count&1)
        if(count&2) h--; else h++;
    }
  return h*3.14159265359/(2*(m-ny)*(n-ny));
}
```

The function counts the number of black pixels in the corners of the filter mask (variable `count`). A single histogram variable `h` is incremented for a `count = 1` and decremented for `count = 3`. An even value of `count` does not change the histogram variable. There are $(m-\nu)*(n-\nu)$ distinct positions of the filter mask. Therefore, `curvature` runs in $O((m-\nu)*(n-\nu))$ time.

7.6 Implementation and Practical Results

The above source code shows the main approach for efficient texture analysis of binary images. All the $3+8\nu$ coefficients can be calculated for each relevant size of the filter mask ν. This program has been implemented on PC Pentium (133 MHz) and optimized for speed. Using the binary image shown in Figure **??** which has 1654 columns and 1136 rows (about two million pixels), the program runs 1.48 sec for $\nu = 1$, 6.42 sec for $\nu = 10$, and 80.23 sec for $\nu = 100$ on this computer.

The most time consuming function is `projection`. We have changed only this function to a table based approach and reduced the run time to 0.71 sec for $\nu = 1$, 2.35 sec for $\nu = 10$, and 19.78 sec for $\nu = 100$.

As an example of application we consider a binary image which is obained from a micro structure of a rolled stainless steel observed in a planar section. Figure **??** shows the typical structure of a micro-duplex structure; in this image the light component is the ferrite phase (α) and the dark one is the austenite phase (γ). The section plane is chosen as a longitutinal micro section of the specimen, i.e. the x-axis is the rolling direction and the y-axis represents the main direction of materials deformation. It is clear the area fraction of α depends on the composition of the steel. For micro-dupex steel the area fraction is about 50 %. A quantity important for process control is the texture of the boundary line between α and γ. It mainly depends on the chosen amount of working as well as on some kind of recovery. The aim is to produce a microstructure with a high specific boundary line length where the specific integral of curvature should be as small as possible.

For the binary image in Figure **??** the estimated area fraction is $\widehat{A_A} = 0.569$. This value has been obtained for $\nu = 0$. Figure **??** shows the texture of the micro structure which is represented by the specific total projection $p_A(\vartheta)$. To demonstrate the influence of a decreasing lateral resolution, $p_A(\vartheta)$ has been estimated for several sizes ν of the applied filter mask. The graphs of the estimates $\widehat{p_A}(\vartheta)$ show that with increasing ν the directional resolution increases such that a rose of intersections is obtained which is practically continuous. Furthermore, this example makes clear that with decreasing lateral resolution the absolute values of $\widehat{p_A}(\vartheta)$ can decrease very fast. Just for $\nu > 3$ this effect is considerable. This is due to the fact that the grain size of the structure is not much greater than the lattice distance Δ. Figure **??** shows the same effect; the specific boundary length decreases for increasing ν. Finally, the texture of the filter mask influences the estimates $\widehat{p_A}(\vartheta)$ by an additional reduction of the lateral resolution for directions close to ϑ_ℓ with $\ell = \nu, 3\nu, 5\nu, 7\nu$. In Figure **??** this effect is very clear for $\nu > 3$. Both effects, the influence of lateral resolution and the influence of the texture of the filter mask on estimates $\widehat{p_A}(\vartheta)$ can be considerably reduced by the methods described in §**??**. Correspondingly modified algorithms are vailable from http://www.informatik.tu-freiberg.de/prof2/publikationen/jom_97.html.

Nevertheless, the diagrams in Figures **??** and **??** have an own meaning; they represent some kind of autocorrelation of the structure.

References

Adams, L. A. (1993) Orientation imaging microscopy: application to the measurement of grain boundary structure. *Materials Science and Engineering* **A166**, 59–66.

Adams, L. A. & D. P. Field (1992) Measurement and representation of grain-boundary texture. *Metallurg. Trans.* **23A**, 2501–2513.

Beneš, V., J. Chadœuf & J. Ohser (1994) On some characteristics of anisotropic fibre processes. *Math. Nachr.* **169**, 5–17.

Chaix, J.-M. & F. Grillon (1996) On the rose of directions measurement on the discrete grid of an automatic image analyser. *J. Microsc.*, **184**, 208–213.

Heijtmans, H. J. A. M. (1994) *Morphological Image Operators.* Academic Press, Boston.

Ohser, J. (1981) A remark on the estimation of the rose of directions of fibre processes. *Math. Operationsforsch. Statist., Ser. Statistics* **12**, 581–585.

Pohlmann, S., J. Mecke & D. Stoyan (1981) Stereological formulas for stationary surface processes. *Math. Operationsforsch. Statist., Ser. Statistics* **12**, 429–440.

Serra, J. (1982) *Image Analysis and Mathematical Morphology.* Academic Press, London.

Stoyan, D. & H. Stoyan (1994) *Fraktale, Formen und Punktfelder.* Akademie Verlag, Berlin.

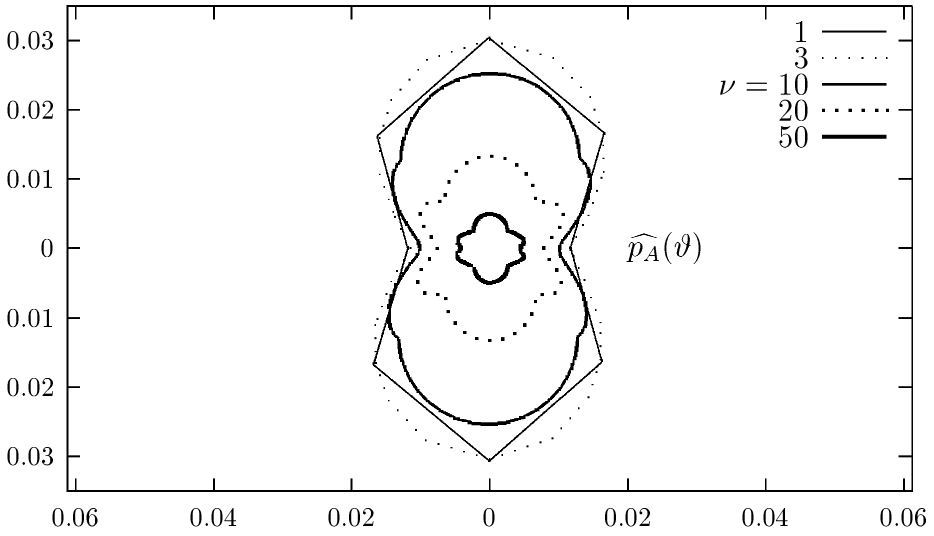

Figure 4: The specific total projection $\widehat{p_A}(\vartheta)$ depending on the order ν of the filter mask.

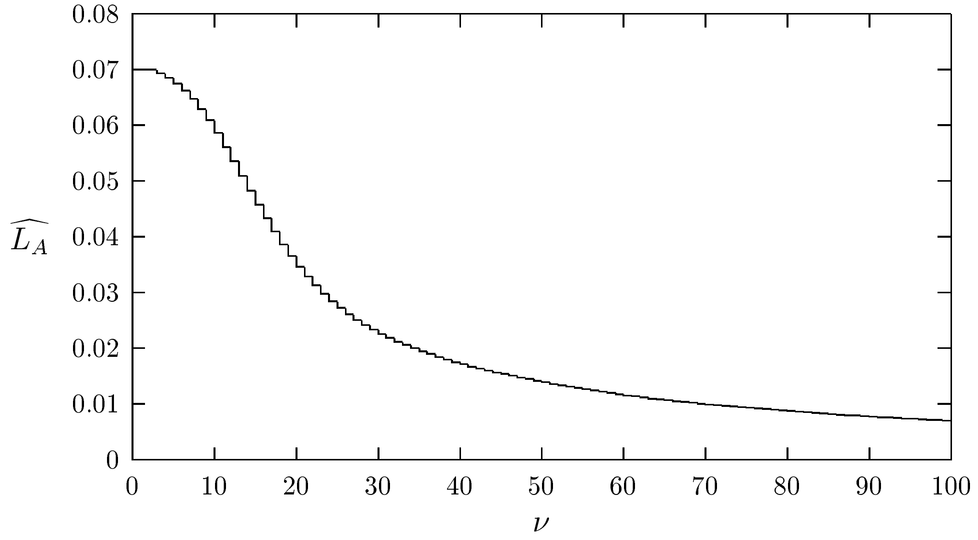

Figure 5: The specific line length $\widehat{L_A}$ depending on the order ν of the filter mask.

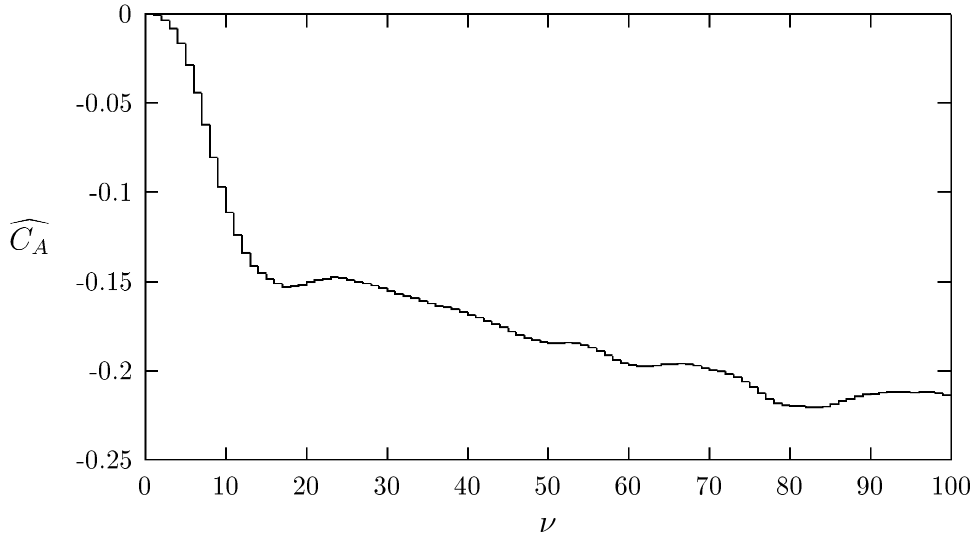

Figure 6: The specific integral of curvature $\widehat{C_A}$ depending on the order ν of the filter mask.