# To the Detection of Non-Metallic Inclusions in Tool Steel Specimens for Polymer Processing Applications

## Diploma Thesis

by

**bakk.techn. Uwe Gallaun**

**Polymer Competence Center Leoben GmbH**
and
**Institut für Kunststoffverarbeitung**
**Montanuniversität Leoben**

Supervisor:
**Dipl.-Ing. Dr.techn. Dieter P. Gruber**
Polymer Competence Center Leoben GmbH

Assessors:
**O.Univ.-Prof. Dipl.-Ing. Dr.techn. Günter Langecker**
**Ass. Prof. Dipl.-Ing. Dr.mont. Walter Friesenbichler**
Institut für Kunststoffverarbeitung - Montanuniversität Leoben

Leoben, November 2008

## Acknowledgement

The research work of this Diploma Thesis was performed within the Framework of the NKP Project ”Charakterisierung von Nichtmetallischen Einschlüssen in Edelstählen”. The PCCL is funded by the Austrian Government and the State Governments of Styria and Upper Austria.

**Special thanks go to:**

Dipl.-Ing. Dr.techn. Dieter P. Gruber
for intensive mentoring, scientific management and supervision of the Diploma Thesis.

O.Univ.-Prof. Dipl.-Ing. Dr.techn. Günter Langecker
and Ass. Prof. Dipl.-Ing. Dr.mont. Walter Friesenbichler
for assessing.

Dipl.-Ing. Hubert Lenger from Böhler Edelstahl GmbH
for making possible to perform the Diploma Thesis.
Dipl.-Ing. Jochen Perko from Böhler Edelstahl GmbH
for an introduction to non-metallic steel inclusions.
Dieter Reppnigg and Ursula Gerhalter from Böhler Edelstahl GmbH
for taking sample images.

**Ein herzliches Dankeschön geht an:**

Meine Eltern Josef und Christa. Danke für Eure Unterstützung.

Meine Familie.

Manuel Haller, Bernd Bernsteiner, Peter Jaunig, Flo und das Ohm Team, Wolfgang Kitzmüller, Markus Zinterl, Gernot Gigerl, Alfred Stadtschnitzer, Sophie Ederer, Fritz Kosir, hellmood entertainment, die B.U.W., U durch I, The Scarves, The Lullabies und fm4.

## Affidavit

I declare in lieu of oath, that i wrote this thesis and performed the associated research myself, using only literature cited in this volume.

Date

Signature

# Abstract

Non-metallic inclusions like oxides and sulfides influence the characteristics of tool steels. After manufacturing the steel quality must be tested by visual observation of polished cut images in order to see if the steel can be used for further production steps. Since the human effort for this kind of quality inspection is very high the object of this work was to investigate whether or not it is possible to extract and classify steel inclusions by automated inspection. Therefore 47 images with a total of 131 inclusions (provided by the Böhler Edelstahl GmbH and Co KG) have been pre-processed, analyzed and classified using different mathematical methodologies. Thereby state of the art operations as well as new developed methodologies have been applied. The classification of sulfides works very well. The separation of oxides turned out to be complex because oxides occur in three different but similar types. Finally, good classification results have been achieved for oxide types as well. As a concluding statement it is to say that a change to automated inspection and classification turned out to be possible. However it is advisable to further improve processing using more sample images of different inclusion types and different types of surface errors like holes and scratches.

# Abstract

Nicht metallische Einschlüsse wie beispielsweise Oxide und Sulfide beeinflussen Eigenschaften und Qualität von Werkzeugstählen. Nach der Stahlerzeugung muss die Qualität anhand von polierten Schliffbildern visuell beurteilt werden, um die weitere Verwendbarkeit der Charge zu ermitteln. Da der menschliche Aufwand für diese Qualitätskontrolle sehr hoch ist, war das Ziel dieser Diplomarbeit, festzustellen ob eine Automatisierung der Überprüfung von Einschlüssen machbar und zweckmäßig ist. Zu diesem Zweck wurden 47 Schliffbilder (zur Verfügung gestellt von Böhler Edelstahl GmbH and Co KG) mit einer Gesamtanzahl von 131 Einschlüssen mit Hilfe von verschiedenen mathematischen Methoden aufbereitet und die Einschlüsse klassifiziert. Dafür wurden sowohl state-of-the-art Methoden als auch neu entwickelte Methodiken angewandt. Die Klassifizierung von Sulfiden funktioniert sehr gut. Die automatisierte Klassifizierung von Oxiden stellte sich als komplex heraus, da Oxide in drei zu unterscheidenden aber ähnlichen Typen auftreten. Letztlich konnten auch für Oxide gute Klassifizierungsergebnisse erzielt werden. Es wurde festgestellt, dass die Automatisierung der Untersuchung und Klassifizierung möglich ist. Zu beachten ist die Ähnlichkeit von Verunreinigungen und Löchern zu den zu klassifizierenden Einschlüssen. Um die Genauigkeit der Klassifizierung weiter zu erhöhen ist die Weiterentwicklung der Analysesoftware mit Hilfe einer größeren Zahl von Schliffbildern ratsam.

# Contents

# 1 Introduction and Motivation

Böhler is one of the most significant companies in the developement and production of special steels, and one of the world's leaders in the areas of high speed steels and tool steels. Different kinds of steel contain inclusions like oxides, sulfides or nitrides that at are responsible for different characteristics. For steel quality type, size and number are crucial aspects.
After manufacturing the cleanliness must be tested, in order to see if the steel can be used for further production steps. Therefore small specimens are grinded, polished and the inclusions are evaluated under a microscope with 100x magnification. The human effort for this inspection is very high because nearly 5000 specimens must be viewed every month. To simplify this process, automated detection has been tested in the past years, but without any conclusions. The object of the feasibility study performed in the framework of the NKP Project "Charakterisierung von Nichtmetallischen Einschlüssen in Edelstählen" was the investigation whether or not it is possible to extract and classify given steel inclusions by automated inspection. It was aimed to conclude with the prognosis if a change to automated inspection and classification is feasible.

In this thesis the processing of images and the results of the classification of non-metallic steel inclusions are described.
The base of the inspection is the DIN 50 602 norm which defines the level of purity. For an automation of this process a camera is placed on a microscope and the provided images shot by this camera were processed and analyzed.

In Section 2 theoretical basis is described. The essential step before different characteristics of an image can be extracted is pre-processing, where mathematical operations transform images to receive convincing data for image analysis. Another important operation are *Co-occurrence Matrices*, which are used to describe images as a composition of elementary structures that carries attributes and have different relations. The aim is to connect these attributes and relations in order to get comprehensive information about image areas that are important for classification.

Section 3 describes how the theoretical basis is applied on sample images of non-metallic inclusions. Standard operations for pre-processing were tested and combined in different ways in order to see if the the different inclusions in the image can be extracted and classified. Therefore two methodologies were used in this work: *Morphological Classification* to get the morphological characteristics and *Classification by Co-occurrence Matrices* to get texture features and type differentiation. MATLAB (The MathWorks, Inc.) with the Image Processing Toolbox was used for image processing, where some features are already prepared and others had to be programed.

In Section 4 the classification of oxide and sulfide inclusions is shown as well as the detection of the shape of the inclusion (line-shaped, dissolved, globular).

Conclusions of this work and an outlook are shown in Section 6.

# 2 Theoretical Basis of Image Processing

The aim of this section is to show the theoretical and mathematical background behind the operations that were used in this work. It is described how the pre-processing of images works with different methodologies like *Filtering* or *Image Segmentation*. Another important part in image analysis are *Co-occurrence Matrices* which are also described in this section.

## 2.1 Pre-processing of Digital Images

Digital images exhibit varying quality levels, which is difficult for later processing. For that reason pre-processing is necessary to get optimal results. Different operations like *Intensity Transformation* or *Filtering* change the image in a way that inclusion recognition and classification can perform better.

### 2.1.1 Spatial Representation of Digital Images

An image may be interpreted as the spatial distribution of the irradiance at a certain area. A computer can't work with continuous analogous images. That is why images are shown as a function of two spatial coordinates $x_1$, $x_2$ for image processing applications which are described in a two-dimensional array of intensity points [1].

$$E(x_1, x_2) = E(\vec{x}) \tag{2.1}$$

These points are called pixels from picture-elements. The positions of the image points are stored in a 2D-array where the first index $m$ denotes the position of the row and the second index $n$ denotes the position of the column.
A digital image consists of $M$ rows and $N$ columns, which are recorded through a $M \times N$ matrix, where the index $m$ runs from 0 to $M - 1$ and the index $n$ from 0 to $N - 1$ (see Fig. 2.1).
The measured irradiance must be transformed into a number of discrete gray values. Usually 256 gray values are used, so that every pixel needs 8 bit storage space.
The pixel that is stored in the matrix represents the average value of irradiance in the corresponding region of the image.

### 2.1.2 Neighborhood Relations

Important features of a matrix are developed by the use of relationship between "neighborhoods", which define connected image areas. Pixels must have adja-

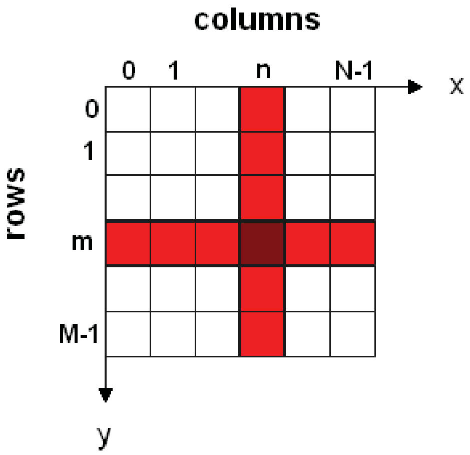


Figure 2.1: Representation of an image by an array with a rectangular grid with $M$ rows and $N$ columns.

cent edges or adjacent corners to belong to the same neighborhood.

In a 2-D array two different kinds of neighborhoods are defined: 4-neighborhood and 8-neighborhood.
Figure 2.2 shows an example how these operations work. The dark red square shows the center pixel of the neighborhood, the brighter red pixels show the neighborhood pixels.
The 4-neighborhood only uses the four pixels in ±0° and ±90° direction in order to describe a connected area (see Fig. 2.2(a)) whereas the 8-neighborhood also uses four pixels in ±45° and ±135° direction (see Fig. 2.2(b)).
In Figure 2.2 (c) the darker region is considered as two regions if a 4-neighborhood is used, but it is considered only as one region if an 8-neighborhood is used.

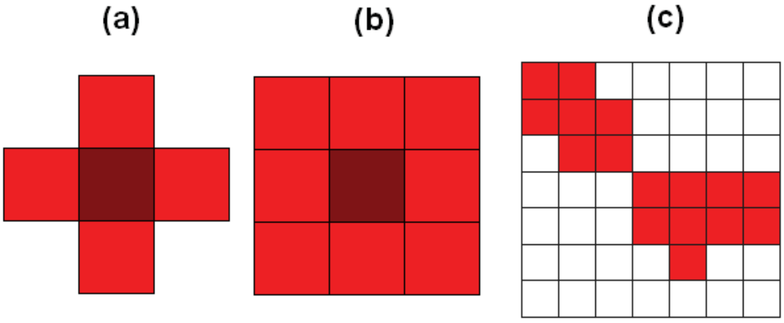


Figure 2.2: Neighborhoods on a rectangular grid: **(a)** 4-neighborhood **(b)** 8-neighborhood **(c)** the colored region shows a connected region in an 8-neighborhood and two regions in a 4-neighborhood [1].

### 2.1.3 Properties of Digital Images

With properties such as distance, slope, angle and coordinate transforms as well as translation, rotation and scaling, properties of different regions in an image can be described.
Therefore a grid vector $\vec{r}$ for the position of each pixel is used to describe features of the matrix [1].

$$\vec{r}_{m,n} = \begin{pmatrix} n\Delta x \\ m\Delta y \end{pmatrix} \tag{2.2}$$

The most frequently used geometrical feature is the *Euclidean Distance* $d_e$ which defines the distance between two points. For a two-dimensional distance the *Euclidean Distance* is better known as the *Pythagorean theorem*. Thereby for distance measurement the *Euclidean Distance* is transfered from continuous space to a discrete grid (see Eq. 2.3)[1].

$$d_e(r, r') = \left\| r - r' \right\| = [(n - n')^2 \Delta x^2 + (m - m')^2 \Delta y^2]^{1/2} \tag{2.3}$$

### 2.1.4 Intensity Transformations

In image processing two important neighborhood processing methodologies are used. The *Intensity* (or *Gray Level*) *Transformations* and the *Spatial Filtering*, which are working directly on the pixels (see Eq. 2.4), where $f(x, y)$ is the input image, $g(x, y)$ is the output image and $T$ is the operator that is applied to $f$ [2].

$$g(x, y) = T[f(x, y)] \tag{2.4}$$

Thereby the first step is to define a rectangular neighborhood around a defined center $(x, y)$ (see Fig. 2.3). The operator $T$ is applied to each location $(x, y)$ to get the output $g$ for which only the pixels in the predefined neighborhood are used. This center runs from pixel to pixel.

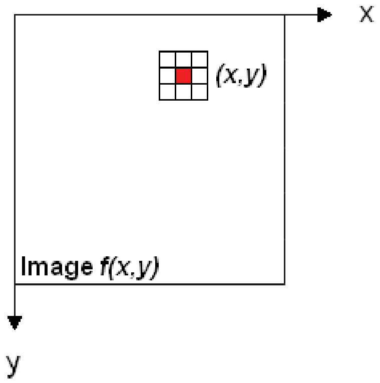


Figure 2.3: A 3 × 3 Neighborhood around a pixel in an image $f(x, y)$ [2].

The simplest transformation is the *Intensity Transformation*, where only one pixel is predefined as neighborhood and the operator $T$ becomes an *Intensity Transformation* function. For this transformation the expression

$$s = T(a) \tag{2.5}$$

is used, in where $a$ is the intensity distribution of the image and $s$ is the intensity distribution of the output matrix [2].

A function that transforms intensity values that are lower than a given threshold value $t$ into a narrow range of darker levels is called *Contrast-Stretching Transformation* (see Eq. 2.6). The levels that are higher are transformed into a narrow range of lighter levels. The exponent $E$ gives the shape of the curve in Fig. 2.4 (a). If the curve is a step function like in Fig. 2.4 (b) the output is a binary image where the values are just 0 and 1 (black and white). This function is called *Threshold Function* [2].

$$s = T(a) = \frac{1}{1 + (t/a)^E} \tag{2.6}$$

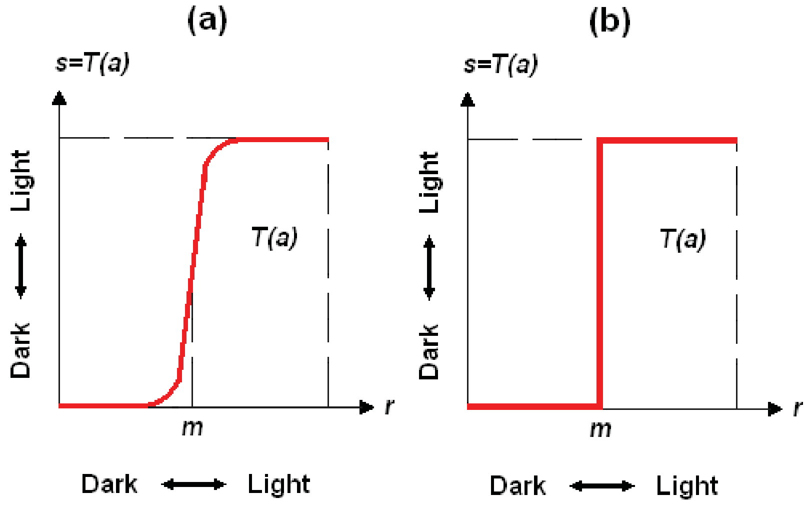


Figure 2.4: Different curve shapes for *Intensity Transformation*: **(a)** *Contrast-Stretching* Transformation **(b)** *Tresholding Transformation* [2].

### 2.1.5 Spatial Filtering

For *Spatial Filtering* an operation is used directly on the pixels of an image [3]. Thereby linear operations multiply each pixel in the neighborhood of a point with a corresponding coefficient in a mask, covering a certain amount of neighborhood pixels (windowing) and process the pixel values of the neighborhood according to a defined rule in order to get the required output (see Fig. 2.5). A mask is a $M \times N$ matrix with predefined coefficients aimed to transform the image specifically.

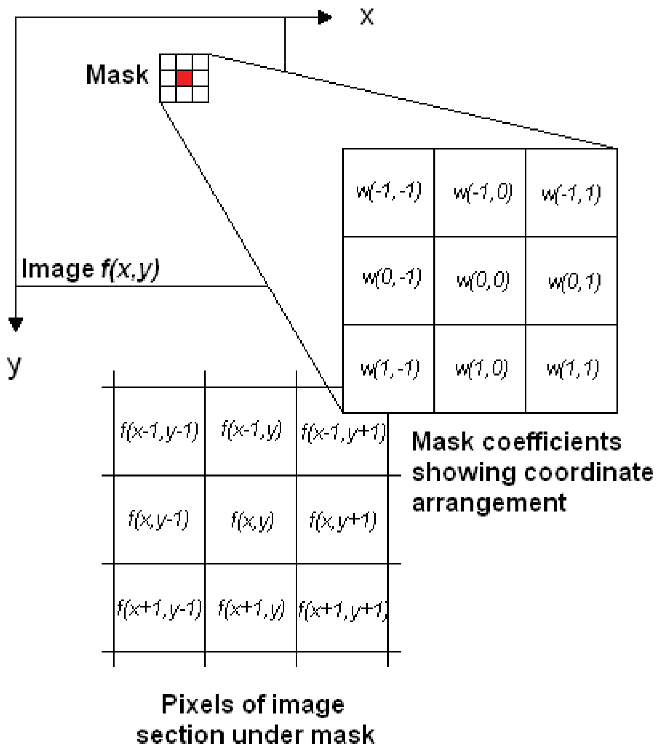


Figure 2.5: *Linear Spatial Filtering*: The figure shows an image with a 3 × 3 mask around a centered pixel. The image is overlapped by the mask [2].

For *Linear Spatial Filtering* two mathematical operations are essential: *Correlation* and *Convolution*.
Figure 2.6 shows the steps of *Correlation* and *Convolution*. Figure 2.6(a) shows the original image $f$ and a mask $w$. Figure 2.6(b) and (c) show how to pad the image with zeros that way that the origin of $f$ is the same as the right bottom point of $w$. The mask is moved from pixel to pixel that way that at least one pixel of the mask overlaps with one pixel of the image. Figure 2.6(d) shows a ”full” *Correlation* where the zeros remain, Fig. 2.6(e) shows the ”same” *Correlation* where the zeros are erased, that the output has the same size as the original image.
*Convolution* is a similar process like *Correlation* but with the mask $w$ rotated by 180°. Figure 2.6(f) shows the padded image overlapped by the rotated mask $w$. Figure 2.6(g) shows a ”full” *Convolution* where the zeros remain, Fig. 2.6(h) shows the ”same” *Convolution* where the zeros are erased, that the output has the same size as the original image.

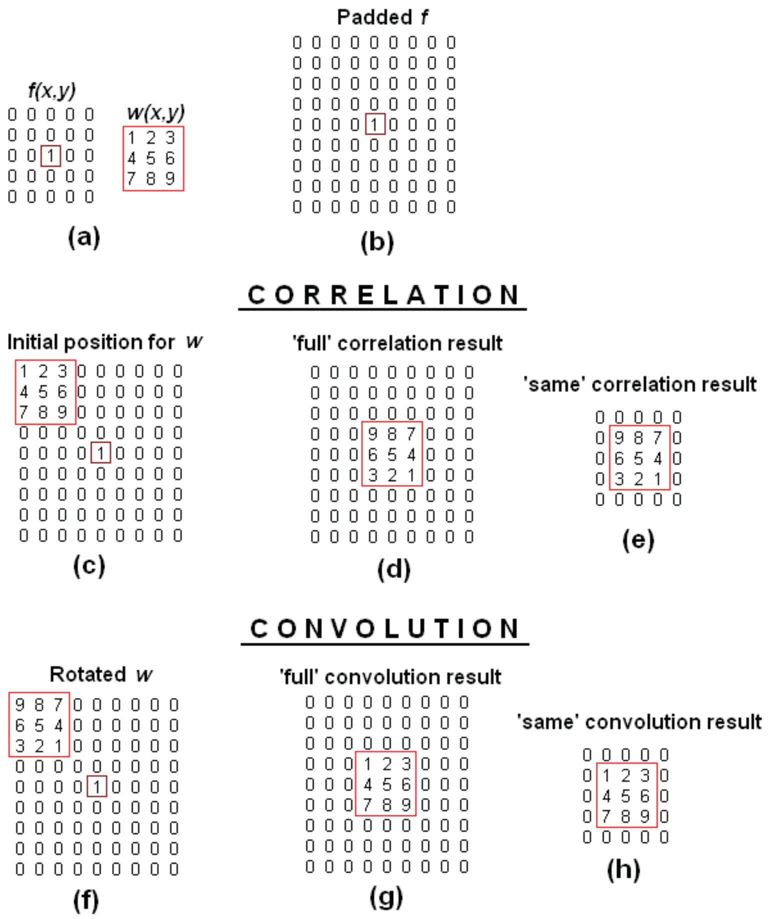

f(x,y)
w(x,y)
(a)
Padded f
(b)
CORRELATION
Initial position for w
(c)
'full' correlation result
(d)
'same' correlation result
(e)
CONVOLUTION
Rotated w
(f)
'full' convolution result
(g)
'same' convolution result
(h)

Figure 2.6: The detailed steps of *Correlation* and *Convolution* as the elementary mathematical operations for a variety of filter processes with an image $f(x, y)$ an a mask $w(x, y)$ [2].

Filters generally used for *Convolution* are:

### Average Filter

The *Average Filter* is a fast method for smoothing a matrix. It reduces the amount of intensity variation between one pixel and adjacent pixels by averaging neighborhood pixels. After applying the *Average Filter* the center pixel in this neighborhood is replaced by the filtered pixel (see Eq. 2.7). $FP$ denotes the filtered pixel, $M \times N$ is the number of pixel in the mask and $p_i$ denotes a single pixel in the mask. The Filter is frequently used to reduce image noise prior further processing.

$$FP = \frac{\sum_{i=1}^{M \times N} p_i}{M \times N} \tag{2.7}$$

### Gaussian Filter

The *Gaussian Filter* is a *Convolution* operator that blurs the image and removes image noise as well. It is similar to the *Average Filter* but it uses a mask that is a crude approximation of the *Gaussian Hump* which is shown with the following mask (see Eq. 2.8 and Eq. 2.9). $\sigma$ denotes the standard deviation of the *Gaussian Distribution*, while $G(x, y)$ denotes the distribution of the pixels after *Filtering* .

$$GM = \begin{bmatrix} 0 & 1 & 2 & 1 & 0 \\ 1 & 4 & 8 & 4 & 1 \\ 2 & 8 & 16 & 8 & 2 \\ 1 & 4 & 8 & 4 & 1 \\ 0 & 1 & 2 & 1 & 0 \end{bmatrix} \tag{2.8}$$

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{\frac{-x^2+y^2}{2\sigma^2}} \tag{2.9}$$

### Laplacian Filter

The *Laplacian Filter* is the second spatial derivative of the image matrix and can be computed by using *Convolution.* It is used for *Edge Detection*, because on edges the gray level transitions have high values and these transitions can be detected very well. The *Laplacian Filter* is very sensitive to noise. So the image is often smoothed with a *Gaussian Filter* at first. $L(x, y)$ denotes the distribution of the pixels after *Filtering.*

$$L(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \tag{2.10}$$

The *Convolution* masks $LM_1$ and $LM_2$ are received by approximating the second derivatives because the input image consists of discrete pixels (see Eq. 2.11).

$$LM_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} ; \; LM_2 \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{2.11}$$

### LoG (Laplacian of Gaussian) Filter

*Convolution* is associative so a *Gaussian Filter* can be convolved with a *Laplacian Filter* first. Instead of using a *Gaussian Filter* and a *Laplacian Filter* separately the *LoG* is used to reduce run-time on the image. $LoG(x, y)$ denotes the distribution of the pixels after filtering.

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[ 1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{\frac{-x^2+y^2}{2\sigma^2}} \tag{2.12}$$

### Prewitt Filter

The *Prewitt Filter* is a *Convolution* operator used for *Edge Detection* and its output is the gradient of the image. *Convolution* is commutative and associative so smoothing can be realized in a convolution mask, because noise is frequently confused with edges. Smoothing works with simple averaging orthogonal to the filter direction. That means for horizontal filters it is smoothed along rows, for vertical filters it is smoothed along columns, which gives rise to the masks $PM_x$ and $PM_y$ (see Eq. 2.13 and Eq. 2.14). The original image $f(x, y)$ is convoluted with the masks to get the resulting matrices $P_x$ and $P_y$ which show horizontal and vertical derivative approximations standing for vertical and horizontal edges (see Eq. 2.15 and Eq. 2.16).

$$PM_x = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \tag{2.13}$$

$$PM_y = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{2.14}$$

$$P_x = \frac{1}{8} PM_x * f(x, y) \tag{2.15}$$

$$P_y = \frac{1}{8} PM_y * f(x, y) \tag{2.16}$$

To receive direction independent information these two results can be combined to the gradient magnitude $P$.

$$P = \sqrt{P_x^2 + P_y^2} \tag{2.17}$$

The gradient direction is found by

$$\Theta = \arctan\left(\frac{P_x}{P_y}\right) \tag{2.18}$$

### Sobel Filter

The *Sobel Filter* is similar to the *Prewitt Filter* but the central row/column of the filter is weighted double.

$$SM_x = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{2.19}$$

$$SM_y = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.20}$$

$$S_x = \frac{1}{8} SM_x * f(x,y) \tag{2.21}$$

$$S_y = \frac{1}{8} SM_y * f(x,y) \tag{2.22}$$

**Unsharp Filter**

The *Unsharp Filter* enhances edges by deducting a smoothed version of an image from the original image. It produces an edge image $U(x,y)$ from the original image $f(x,y)$.

$$U(x,y) = f(x,y) - f_{smooth}(x,y) \tag{2.23}$$

### 2.1.6 Morphological Operators

*Morphological Operators* use a a structuring element (a defined matrix of zeros and ones) to change an object in an image in a way that makes possible to get morphological information like area, width, axes length or diameter. To receive information about these aspects images are transformed to binary images. That means the images only exist of two intensity values (0 and 1) and are generated by a threshold operation (see Section 2.1.4). The most important *Morphological Operations* are *Dilation* and *Erosion*. *Dilation* can be used to fill holes in connected regions and *Erosion* can be used to reduce noise.

**Dilation**

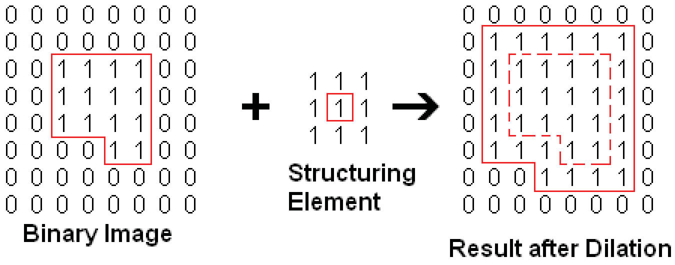
Dilation is an operation where an object in a binary image grows, that is to say the boundaries of an object are enlarged. A structuring element is overlapped on a binary image.

The structuring element and the binary image consists only of 0 and 1. If the center of the structuring element meets a 1 of the binary image the other structuring element values are transfered to the binary image. The values of the binary image are now identical to the values of the structuring element - boundaries are ”growing” (see Fig. 2.7). With this operation holes in a connected region can be filled.

*Dilation* is commutative. That means it does not matter if the binary image is used as structuring element or vice versa.

The mathematical description is given by Equation 2.24 where $G$ is a set of pixels with value 1 in the matrix and and $M$ are non-zero mask pixels. $M_p$ is the mask shifted with the center to the pixel $p$ [1].

$$G \oplus M = \{p : M_p \cap G \neq \oslash\} \tag{2.24}$$



```
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 1 1 1 1 0 0
0 0 1 1 1 1 0 0
0 0 1 1 1 1 0 0
0 0 0 0 1 1 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

```
1 1 1
1 1 1
1 1 1
```

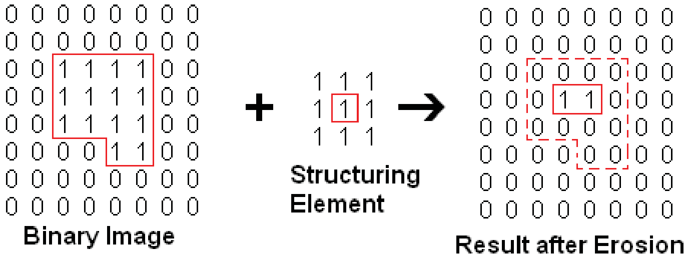
```
0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 0
0 1 1 1 1 1 1 0
0 1 1 1 1 1 1 0
0 1 1 1 1 1 1 0
0 1 1 1 1 1 1 0
0 0 0 1 1 1 1 0
0 0 0 0 0 0 0 0
```

Figure 2.7: *Dilation*: A binary image with a rectangular object (red) is dilated with a 3×3 matrix as structuring element, where the red box denotes the center. As a result the object enlarges. The dashed line shows the original object.

### Erosion

*Erosion* is the inverse operation to *Dilation* and shrinks an object in a binary image. Like *Dilation* a structuring element is overlapped on the image but in the case of *Erosion* only the points that are identical to the structuring element get the value one. With this operation noise can be reduced. The mathematical description is given by Equation 2.25 [1].

$$G \ominus M = \{p : M_p \subseteq G\} \tag{2.25}$$



```
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 1 1 1 1 0 0
0 0 1 1 1 1 0 0
0 0 1 1 1 1 0 0
0 0 0 0 1 1 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

```
1 1 1
1 1 1
1 1 1
```

```
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 1 1 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

Figure 2.8: *Erosion*: A binary image with a rectangular object is eroded with a $3 \times 3$ matrix as structuring element, where the red box denotes the center. As a result the object shrinks. The dashed line shows the original object.

*Erosion* followed by *Dilation* is called *Morphological Opening*, *Dilation* followed by *Erosion* is called *Morphological Closing*.

### 2.1.7 Image Segmentation

*Image Segmentation* generates connected regions with common characteristics such as color, intensity or texture in order to receive features of these regions. Most important usage of *Segmentation* is in the classification of medical images. Sutton and Hall [4] for example used texture features for classification of pulmonary diseases or Harms et al. [5] used a combination of texture and color features to diagnose leukemic malignancy. Other usages are document processing like classification of newspaper image blocks [6], [7], face [8], [9] and fingerprint [10], [11] recognition, analysis of satellite images [12], [13], machine vision [14], [15] or automatic traffic controlling systems [16], [17].

For *Segmentation* mainly two features are of interest: Discontinuity and Similarities. At Discontinuities the intensity value changes abrupt. That is the case with edges for example. In another case similarities are used to define connected areas.

To detect points, lines or edges a mask (similar to *Filtering*) runs from pixel to pixel in the image. A $N \times M$ mask is used, $w$ denotes the intensity value and $z$ denotes the mask coefficient.

$$R = \sum_{i=1}^{N \times M} w_i \cdot z_i \tag{2.26}$$

In this section methods for detecting points, lines and edges are presented. These methods are used to segment a pre-processed digital image to get information about different regions.

#### Point Detection

If a pixel is tested whether if it is part of a defined segment or not, that operations are called threshold operations.

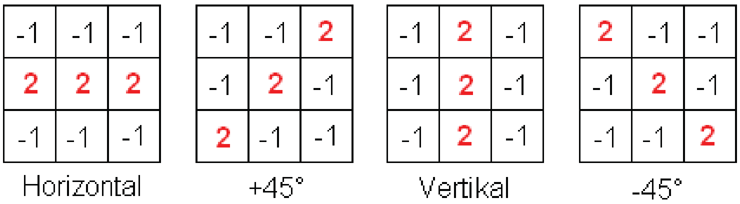
The mask shown in Fig. 2.9 is overlapped with an image and a single point is found if $|R| \geq T$. $T$ is a positive Threshold. In Fig. 2.9 8 was chosen as centerpoint value because in continuous regions the return is 0. In other words, the sum of all coefficients of the matrix is 0.

| | | |
|---|---|---|
| -1 | -1 | -1 |
| -1 | 8 | -1 |
| -1 | -1 | -1 |

Figure 2.9: Example of a mask for Point Detection.

### Line Detection

To detect lines in 0°, +45°, 90° and -45° direction masks like shown in Fig. 2.10 are used, where the red values indicate the origins. If the mask origin overlays points of the same value in the image these pixels are detected as linepoints. There are four direction depended results and the direction with the highest value defines the direction of the linepoint.



| | | |
|---|---|---|
| -1 | -1 | -1 |
| 2 | 2 | 2 |
| -1 | -1 | -1 |

Horizontal

| | | |
|---|---|---|
| -1 | -1 | 2 |
| -1 | 2 | -1 |
| 2 | -1 | -1 |

+45°

| | | |
|---|---|---|
| -1 | 2 | -1 |
| -1 | 2 | -1 |
| -1 | 2 | -1 |

Vertikal

| | | |
|---|---|---|
| 2 | -1 | -1 |
| -1 | 2 | -1 |
| -1 | -1 | 2 |

-45°

Figure 2.10: Examples of masks for Line Detection.

If only a certain orientation is needed the mask dedicated for this orientation is used.

### Edge Detection

For *Edge Detection* the first and the second derivate of the image matrix are needed, as an edge means a significant change of local gray level values. The derivatives are defined as follows (Eq. 2.27 and Eq. 2.28):

$$\nabla f = \begin{pmatrix} G_x \\ G_y \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix} \tag{2.27}$$

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \tag{2.28}$$

The absolute value of $\nabla f$ is

$$G = mag(\nabla f) = [G_x^2 + G_y^2]^{1/2} = [(\partial f/\partial x)^2 + (\partial f/\partial y)^2]^{1/2} \tag{2.29}$$

The angle where the maximum occurs is

$$\Theta(x,y) = \arctan\left(\frac{G_x}{G_y}\right) \tag{2.30}$$

*Steps of edge detection*

Locations where the gradients $G_x$ and $G_y$ are bigger than a defined threshold and locations where divergence is zero are searched by the use of convolution masks. An inflection point is shown by a vanishing divergence and the gradient shows the slope of the curve. If the slope is bigger than a defined threshold and it is an inflection point the point is detected as edge point.
Figure 2.11 shows two examples: The black dots correspond to pixels with different gray level values. The left curve has a gray level transition that is detected as edge point (green cross), the right curve has no edge point because the slope is smaller than the threshold even though the point is an inflection point (red cross).

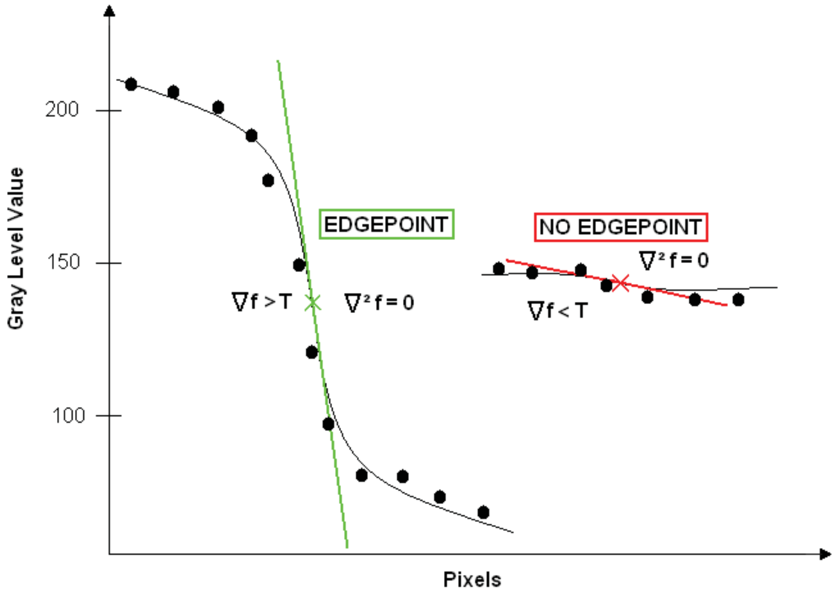
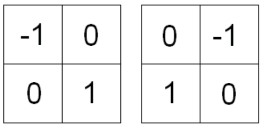


Figure 2.11: A cut through an image where the black dots correspond to pixels with different gray level values. The left curve has a gray level transition that is detected as edge point (green cross), the right curve has no edge point because the slope is smaller than the threshold even though the point is an inflection point (red cross).

*Masks for edge detection*

For edge detection different convolution masks are used. The most common masks are *Sobel, Prewitt, LoG, Roberts* and *Canny*. *Sobel, Prewitt* and *LoG* are discussed in Section 2.1.5. A *Roberts Filter* uses the masks in Fig. 2.12 and it is the simplest mask for edge detection.



| | |
|---|---|
| -1 | 0 |
| 0 | 1 |

| | |
|---|---|
| 0 | -1 |
| 1 | 0 |

Figure 2.12: The *Roberts Masks*.

The most used edge detection mask is the *Canny* edge detector, developed 1986 by John F. Canny [18]. It consists of a few operations: First the image is convolved with a *Gaussian Filter* (to reduce noise) and with a *Sobel Filter* (*Prewitt* or *Roberts* are also possible) to get $G$ and $\Theta$ (see Eq. 2.29 and Eq. 2.30). If a point has its local maximum in the gradient direction, it is detected as an edgepoint. To ensure that the edge is only 1 pixel wide, $G(x, y)$ of every adjacent pixels in the neighborhood is tested. If one pixel has a higher $G(x, y)$ than the edgepoint, the pixel will be set to 0, except the pixel is on the computed direction that means it is an edgepoint too. Last step is to define the thickness of the edge to avoid a cracking up. For this purpose an operation called *Hysterese* is used: Therefore two thresholds $T1$ and $T2$ are used, while $T1 < T2$. The image matrix is scanned until a pixel has a greater value than $T2$ (the origin of the edge). This edge is scanned in both directions and every pixel with a value greater than $T1$ is detected as part of the edge.

## 2.2 Co-occurrence Matrices

An image can be described as a composition of elementary structures that carries attributes and have relations. Attributes are features like the gray level value or the gradient. Relations are features like the distance between two pixels.

In the object recognition field the most important attributes are textures and gray levels. They affect how an image is perceived and they always appear together whereat one feature can dominate. The aim is to connect these features in order to get comprehensive information about different areas in an image. In the beginning of texture analysis autocorrelation functions [19], power spectra [20], restricted first- and second order Markov mashes [21] and relative frequencies of various gray levels [12] have been employed. For this purpose R. Haralick developed the method of *Co-occurrence Matrices*, where texture information is stored in a spatial array[22].

Descriptors that contain information about the texture characteristics are frequently used operations to analyze textures in object recognition [23], [24]. This operations are used for example in medicine [25], [26], to detect tumors [27], [28], for the analysis of satellite images [29], [30] or for volumetric data analysis [31].

To create a *Co-occurrence Matrix* a rectangular image matrix of $M \times N$ pixels is taken. The occurrence structures can be stored in matrices with the frequency values $P_{\phi,d}(i,j)$, showing how often two pixels with a defined distance $d$ from each other and a defined gray level value combination occur in the image matrix. Four directions are defined (0°, 45°,90°,135°) in which the neighbor with the distance $d$ and the given gray level combination is found.

In the general form frequencies can be written as [22]:

$$\begin{gathered} P_{0,d} = |((k,l),(m,n)) \in (M \times N) \times (M \times N) : \\ k-m=0, |l-n| = d, f(k,l) = i, f(m,n) = j| \end{gathered} \tag{2.31}$$

$$\begin{gathered} P_{45,d} = |((k,l),(m,n)) \in (M \times N) \times (M \times N) : \\ (k-m=d, l-n=-d) OR (k-m=-d, l-n=d), f(k,l) = i, f(m,n) = j| \end{gathered} \tag{2.32}$$

$$\begin{gathered} P_{90,d} = |((k,l),(m,n)) \in (M \times N) \times (M \times N) : \\ |k-m| = d, l-n=0, f(k,l) = i, f(m,n) = j| \end{gathered} \tag{2.33}$$

$$\begin{gathered} P_{135,d} = |((k,l),(m,n)) \in (M \times N) \times (M \times N) : \\ (k-m=d, l-n=d) OR (k-m=-d, l-n=-d), f(k,l) = i, f(m,n) = j| \end{gathered} \tag{2.34}$$

In Fig. 2.13 the values 1 and 5 have the smallest distance in direction 0°, 8 and 4 in direction 45°, 7 and 3 in direction 90° and 6 and 2 in direction 135°.

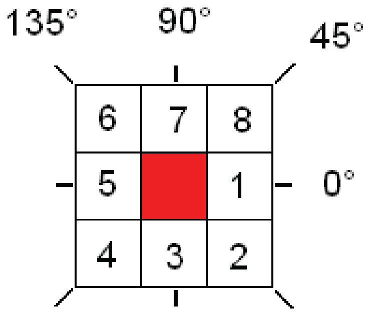


Figure 2.13: The four different directions which are used in calculations with co-occurrence matrices.

To show how *Co-occurrence Matrices* are generated an example image is used (Fig. 2.14 (a)). It is a $4 \times 4$ pixel image with 4 gray levels from 0 to 3 (Fig. 2.14 (b)). Fig. 2.14 (c) shows the general form of any gray level spatial dependence matrix. That means the position $(3, 2)$ describes how frequent two pixels with the gray level values 3 and 2 occur in the defined direction and distance. In the given example distance 1 is chosen.

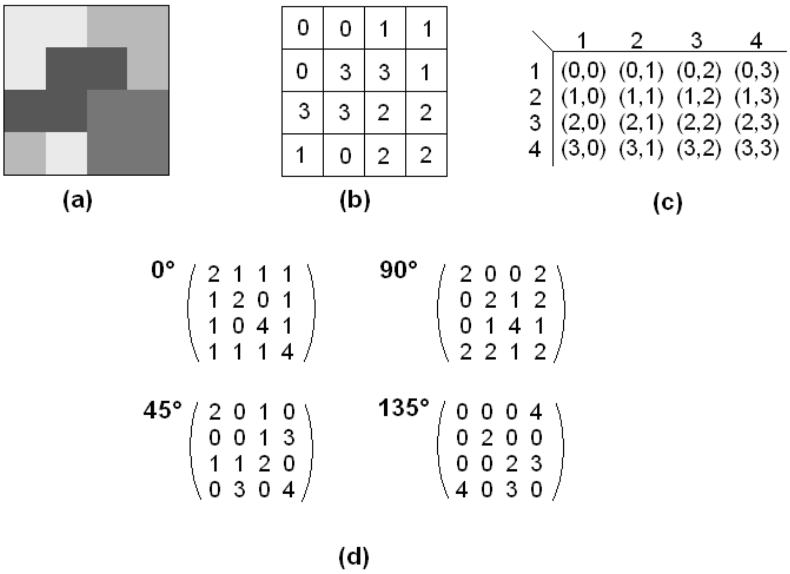
The gray level spatial dependence matrix is generated as follows. Element $P_{0,1}(0, 0)$ shows how often two pixels with distance 1 and the gray level value 0 occur in direction 0°. In the example this occurs two times (Fig. 2.14(d)). Element $P_{0,1}(3, 2)$ shows how often two pixels with distance 1 and the gray level values 3 and 2 occur in direction 0° - that occurs one time. The matrix is symmetric so that $P_{0,1}(2, 3)$ equals $P_{0,1}(3, 2)$.

### 2.2.1 Textural Features for Image Classification

One challenge in the field of image processing is to select adequate descriptors of the objects to be found in the image and represent them adequately in a mathematical formulation. A lot of different types of descriptors were defined in literature like shape descriptors [32] and region descriptors [33]. For the representation of features like semantic nets [34], Freeman chain codes [35], gray level run length matrices [36] or *Co-occurrence Matrices* [22] a lot of descriptors were defined too.

Haralick et al. defined 14 measures of textural features (see Tab. 2.1) that are based on the concept of *Co-occurrence Matrices*. This is adequate since all information of the textural features is stored in the spatial gray level distribution matrices [22].

In the following the features $f1$, $f2$ and $f3$ are discussed in detail, because these features represent visual measures [37], [38]. They are descriptors for the homogeneity, contrast and gray tone linear dependencies of the image and give conclusions about how a viewer appreciate these texture characteristics. That's important for inclusion detection, which is needed in this work, because inclu-



|   |   |   |   |
|---|---|---|---|
| 0 | 0 | 1 | 1 |
| 0 | 3 | 3 | 1 |
| 3 | 3 | 2 | 2 |
| 1 | 0 | 2 | 2 |

|   | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | (0,0) | (0,1) | (0,2) | (0,3) |
| 2 | (1,0) | (1,1) | (1,2) | (1,3) |
| 3 | (2,0) | (2,1) | (2,2) | (2,3) |
| 4 | (3,0) | (3,1) | (3,2) | (3,3) |

$$0° \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 4 & 1 \\ 1 & 1 & 1 & 4 \end{pmatrix} \quad 90° \begin{pmatrix} 2 & 0 & 0 & 2 \\ 0 & 2 & 1 & 2 \\ 0 & 1 & 4 & 1 \\ 2 & 2 & 1 & 2 \end{pmatrix}$$

$$45° \begin{pmatrix} 2 & 0 & 1 & 0 \\ 0 & 0 & 1 & 3 \\ 1 & 1 & 2 & 0 \\ 0 & 3 & 0 & 4 \end{pmatrix} \quad 135° \begin{pmatrix} 0 & 0 & 0 & 4 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 3 \\ 4 & 0 & 3 & 0 \end{pmatrix}$$

Figure 2.14: **(a)** Gray level image **(b)** Gray level matrix **(c)** General form of a gray level spatial dependence matrix **(d)** Co-occurrence matrices for different directions.

sions are commonly viewed by visual observers and then rated as image error or inclusion [22].

Table 2.1: Haralick Classificators [22].

| | Visual Measures |
|---|---|
| f1 | Angular Second Moment (ASM) |
| f2 | Contrast |
| f3 | Correlation |
| | Statistical Measures |
| f4 | Variance |
| f5 | Inverse Difference Moment-Homogeneity |
| f6 | Sum Average |
| f7 | Sum Variance |
| f10 | Difference Variance |
| | Measures that are based on Information Theory |
| f8 | Sum Entropy |
| f9 | Entropy |
| f11 | Difference Entropy |
| | Measures that are based on Correlation |
| f12, f13 | Information Measures of Correlation |
| f14 | Maximal Correlation Coefficient |

### Angular Second Moment - ASM

The *Angular Second Moment* provides information about the homogeneity of the image (see Eq. 2.35).

$$f1 = \sum_i \sum_j P_d^2(i,j) \tag{2.35}$$

$P_d(i,j)\ldots$ Spatial gray level distribution matrix

If there are few gray level transitions, which means the image is homogeneous, there will be fewer values with large magnitude in the matrix $f1$. If the image has more transitions there will be more small values and so the average ASM will also be smaller. Figure 2.15 shows two cutouts of an image from a steel-specimen. Figure 2.15 (a) shows an inclusion in the specimen, Fig. 2.15 (b) is nearly homogeneous. In Fig. 2.15 (a) there are more transitions than in Fig. 2.15 (b), so the ASM values for Fig. 2.15 (b) are higher than for Fig. 2.15 (a) (see Tab. 2.2 and Tab. 2.3).

### Contrast

The *Contrast* Feature is the difference moment and provides information about local gray value changes in the image (see Eq. 2.36).

$$f2 = \sum_i \sum_j (i-j)^2 P_d(i,j) \tag{2.36}$$

The more variations occur in the image, the higher the *Contrast* value will be.

In Fig. 2.15 (a) there is more *Contrast* than in Fig. 2.15 (b), so the *Contrast* values for Fig. 2.15 (b) are lower than for Fig. 2.15 (a) (see Tab. 2.2 and Tab. 2.3).

### Correlation

The *Correlation* feature measures the gray level dependence of the image (see Eq. 2.36).

$$f3 = \frac{\sum_i \sum_j (i-\mu_x)(j-\mu_y)P_d(i,j)}{\sigma_x \sigma_y} \tag{2.37}$$

$$\mu_x, \mu_y \ldots \text{Means of } P_d$$
$$\sigma_x, \sigma_y \ldots \text{Standard deviation of } P_d$$

In Fig. 2.15 (a) the *Correlation* value is higher than in Fig. 2.15 (b), because the image without inclusion is nearly homogeneous plus some additive noise. The noise is mostly uncorrelated so the *Correlation* values are lower than in the image with the inclusion (see Tab. 2.2 and Tab. 2.3).

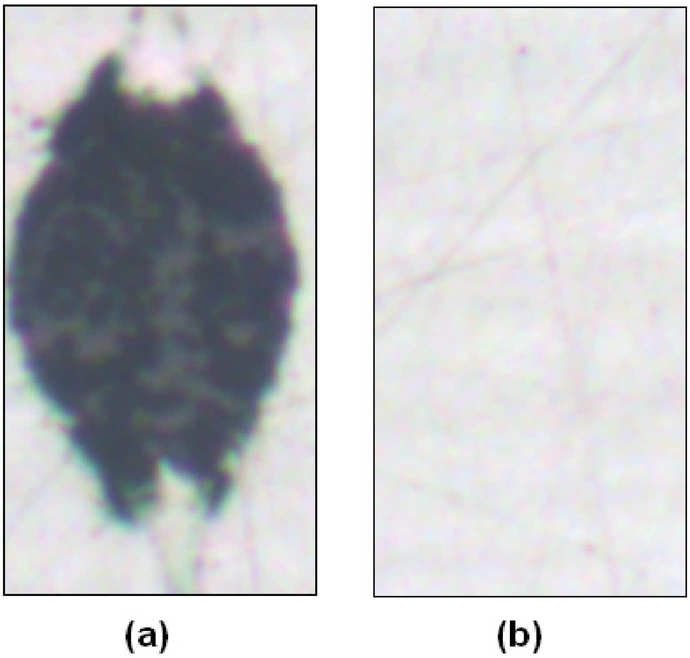


Figure 2.15: Examples on which the Haralick features were tested: **(a)** example with inclusion **(b)** homogeneous Example without inclusion.

Table 2.2: Haralick Classificators for image19 (a).

| Angle | ASM | Contrast | Correlation |
|---|---|---|---|
| 0° | 0.1223 | 0.6181 | 0.9730 |
| 45° | 0.1122 | 0.8269 | 0.9639 |
| 90° | 0.1327 | 0.3504 | 0.9848 |
| 135° | 0.1131 | 0.7983 | 0.9651 |
| Avg. | 0.1201 | 0.6484 | 0.9717 |

Table 2.3: Haralick Classificators for image19 (b).

| Angle | ASM | Contrast | Correlation |
|---|---|---|---|
| 0° | 0.2719 | 0.3744 | 0.6037 |
| 45° | 0.2527 | 0.4516 | 0.5168 |
| 90° | 0.2696 | 0.3254 | 0.6790 |
| 135° | 0.2393 | 0.5141 | 0.4500 |
| Avg. | 0.2584 | 0.4163 | 0.5623 |

### 2.2.2 Multidimensional Co-occurrence Matrices

A multidimensional *Co-occurrence Matrix* is an $M$-dimensional array with elements that have the form $w(a_1, a_2..., a_{m1}; b_1, b_2..., b_{m2})$. Parameter $a$ stands for a certain attribute value, for example the gray level value. $b$ covers the values of relations, for example the *Euclidean Distance*. With this method the relationship between two attributes can be precisely described [39], [40].
The gray values $a_1(i)$ and $a_2(j)$ of two pixels and their Euclidean difference $b_1(i, j)$ are taken to get a matrix $w(a_1(i), a_2(j), b_1(i, j))$. That means for example that a matrix $w(190, 200, 10)$ describes two pixels with the gray level values 190 and 200 and the *Euclidean Distance* 10. The amount of occurrence is stored in the $M$-dimensional array.

# 3 Methodologies for the Classification of Non-Metallic Steel Inclusions

## 3.1 Introduction

Böhler Edelstahl GmbH and Co KG produces a wide range of steel grades with different material properties. The type, the size and the number of inclusions are responsible for these properties. Before steel can be go through further processing it has to pass a strict quality control. Therefore small specimens are grinded, polished and the inclusions are evaluated under a microscope with 100x magnification. A clip of $12 \times 18$ $mm$ of the specimen is subdivided into 200 visual fields. The occurrence of inclusions is checked for every visual field and the results are evaluated based on DIN 50 602.
DIN 50 602 defines the level of purity. It is a specification of non-metallic inclusions in steel specimens in form of oxides and sulfides. For declaring the inclusions a board with different types and sizes of inclusions is used (Figure 3.1). The board is structured in 10 columns and 9 rows. Column 0 and 1 characterize line-shaped sulfides (strichförmige Sulfide - SS), where column 0 stands for thinner and column 1 stands for thicker line-shaped inclusions. Column 2 to 4 characterize dissolved oxides (aufgelöste Oxide - OA), column 5 to 7 characterize line-shaped oxides (strichförmige Sulfide - OS) and column 8 to 9 characterize globular oxides (globulare Oxide - OG). Nine images in a column (with size index 0 to 8) range from the smallest to the biggest inclusion while doubling the expanse of inclusions from row to row. Inclusions are viewed and measured under a microscope with 100x magnification. The range at the lower right of each image (f.e. $>$ 4-6) are the minimum and maximum inclusion length for this size class (f.e. OS 5.0). The unit is $\mu m * 10^{-1}$ so that the values from the board equal 10 percent of the real size.

For the analysis two methodologies are specified in DIN 50 602:

Methodology M registrates only the maximum sized inclusion of each inclusion type. For a specimen set of the same steel, the size index for each inclusion type is averaged. The cut-area is 200 $mm^2$.

Methodology K registrates every inclusion. The number of inclusions of every type and every size index is stored and multiplied with different significance factors. The results are summarized and transformed for an area of 1000 $mm^2$. The cut-area is 100 $mm^2$.

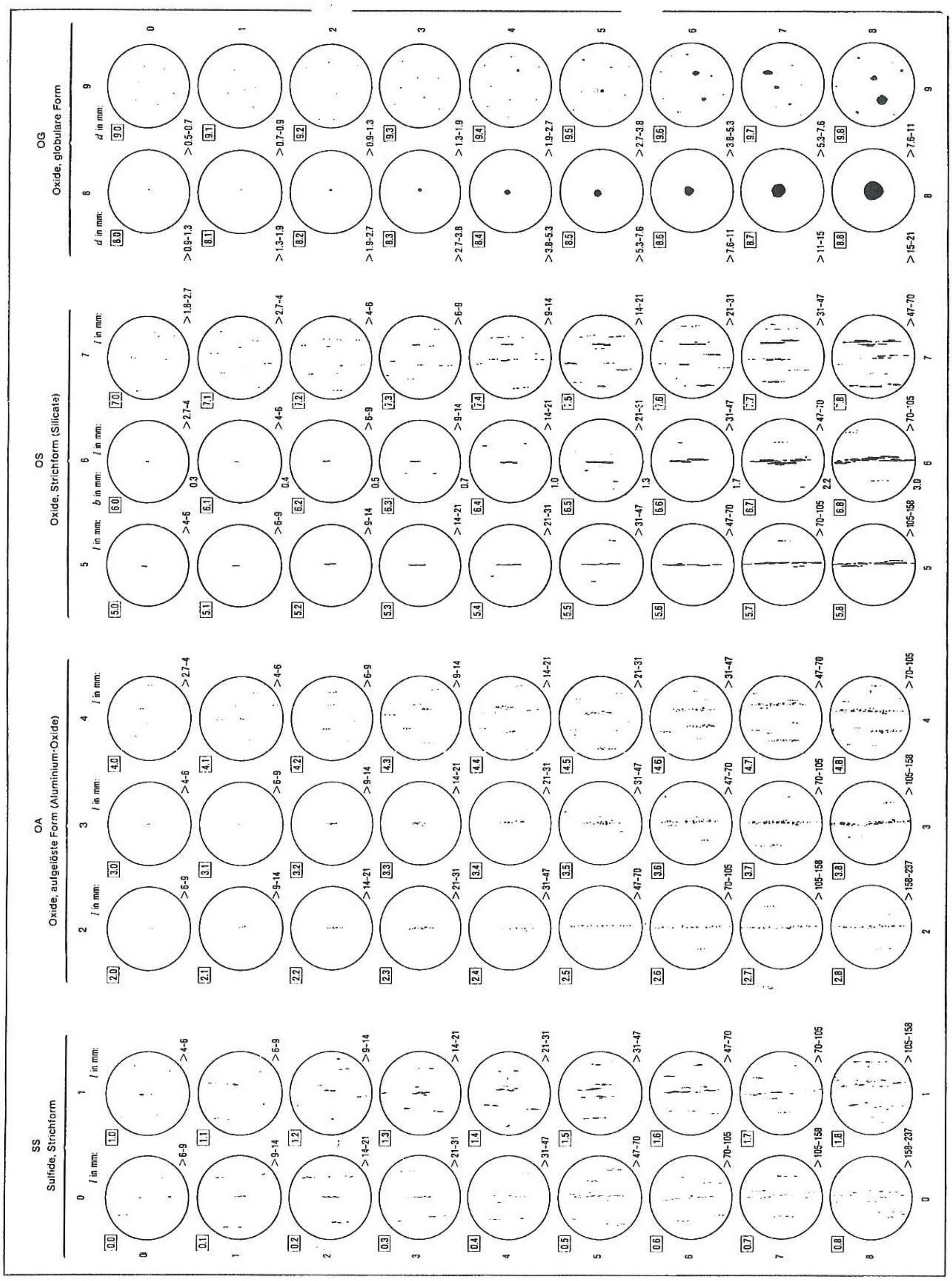


Figure 3.1: DIN 50 602: Board with images of different inclusions. The columns define the type of inclusion: Dissolved Oxides (OA), Line-shaped Oxides (OS), Globular Oxides (OG) and Line-shaped Sulfides (SS). The rows range from the smallest to the biggest inclusion. The range at the lower right of each image defines the size class. The unit is $\mu m * 10^{-1}$ so that the values from the board equal 10 percent of the real size.

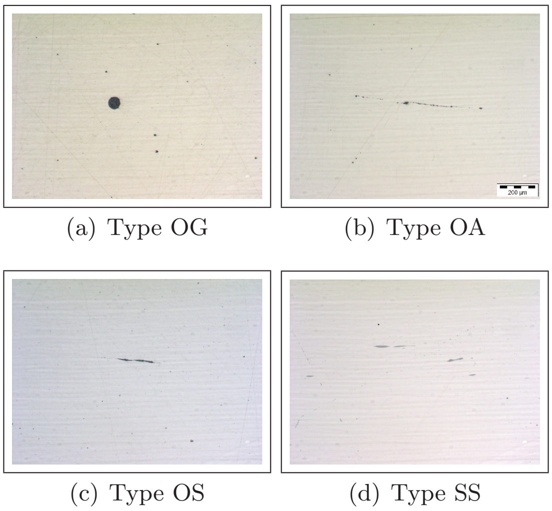
With the results of the DIN 50 602 analysis it is determined whether the requirements to create the necessary material properties are met or not. For an automation of this process a camera is placed on the microscope. In this work the provided images that were shot by this camera were processed and analyzed in order to make a point if a change to full automation is feasible. The image size was 1544 × 2080 pixels and the file format was $TIF$.
This section is aimed to show which methods were used for processing and analyzing of the images.

## 3.2 Pre-processing of Sample Images with Non-Metallic Inclusions

Pre-processing is the essential step before different characteristics of an image can be extracted. With mathematical operations matrices where the image information is stored are transformed to receive convincing data for image analysis. In this work standard operations for pre-processing were tested and combined in different ways to get expressive results. The used software was MATLAB (The MathWorks, Inc.), with the Image Processing Toolbox, where some features are already prepared and others had to be programed.
This section describes the following operations for improving the image information:

- *Intensity Transformation* is used to enhance separation of objects from the background.
- *Spatial Filtering* is used to eliminated noise or scratches. Different filters are used.
- *Thresholding* turns the image into a binary image.
- *Dilation* and *Erosion* transform the shape of an object to erode noise or to fill holes.
- *Edge Detection* is used to define the edges of connected regions.
- *Regiongrow* serves to find connected pixels that belong to a larger region.

Figure 3.2 shows an example of each specimen type on which these operations were used.



(a) Type OG

(b) Type OA

(c) Type OS

(d) Type SS

Figure 3.2: Images (by Böhler Edelstahl GmbH and Co KG and Co KG) of four different types of inclusions.

### 3.2.1 Intensity Transformation

The first pre-processing step is to change the intensity in order to seperate dark objects (inclusions) from the bright background (steel matrix) which exhibits scratches and noise (for theory see Section 2.1.4).
The feature *Imadjust* was used for *Intensity Transformation* and there are five input values to modify:
Low Input and High Input are mapped to Low Output and High Output. This means that only values between Low Input and High Input are relevant. The rest is cut off. The values below Low Input are transformed to Low Output and the values above High Input are transformed to High Output (Fig. 3.3). Figure 3.4 shows three test calculations with different input and output values. The gamma value defines the shape of the curve (Fig. 3.3) between Low Input and High Input, which declares if the output image is brighter (gamma<1, Fig. 3.5(a)) or darker (gamma>1, Fig. 3.5(b)) as the input image. If gamma is 1, the intensity won't change (Fig. 3.5(c)).

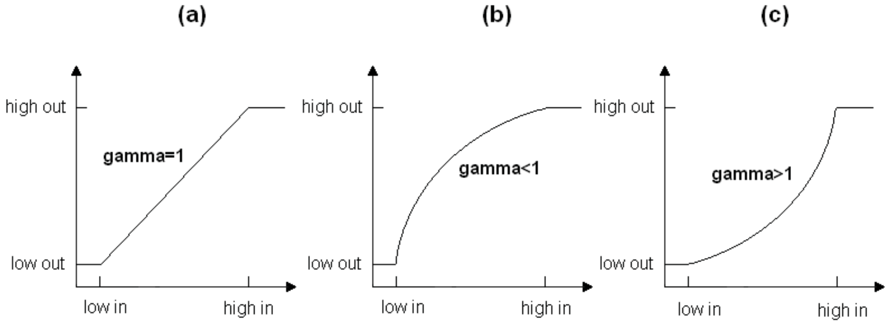
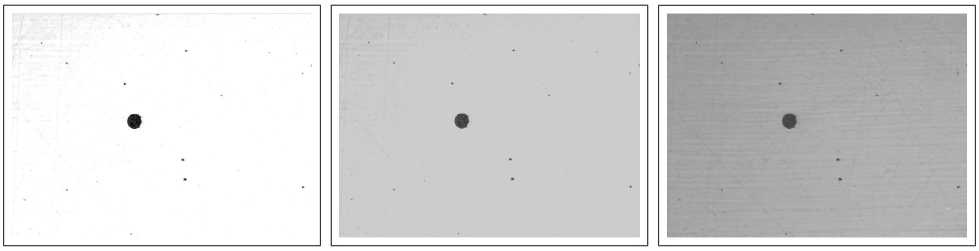


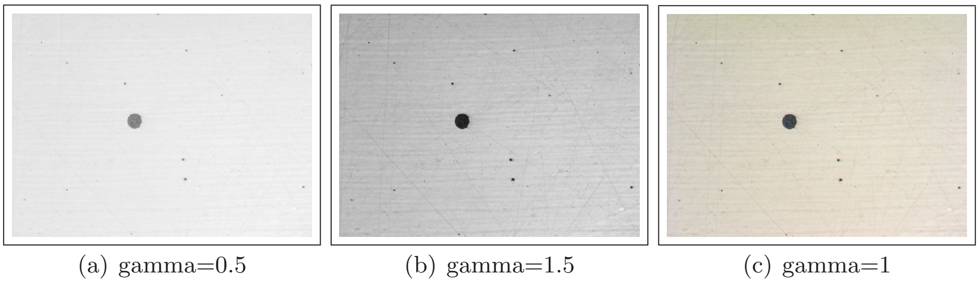
Figure 3.3: *Intensity Transformation* with different gamma values.



(a) In:0.2;0.8 Out:0;1 (b) In:0.2;0.8 Out:0.2;0.8 (c) In:0;1 Out:0.2;0.8

Figure 3.4: *Intensity Transformation* with different in- (In:low;high) and outputs (Out:low;high) and gamma=1.

Values shown in Tab. 3.1 turned out to be best and have been chosen for further processing. Figure 3.6 shows the resulting image.



(a) gamma=0.5 (b) gamma=1.5 (c) gamma=1

Figure 3.5: *Intensity Transformation* with different gamma values and in- and output 0 and 1.

Table 3.1: Settings for MATLAB-function *Imadjust*.

| | |
|---|---|
| Low Input | 0.25 |
| High Input | 0.70 |
| Low Output | 0.00 |
| High Output | 1.00 |
| Gamma | 0.80 |

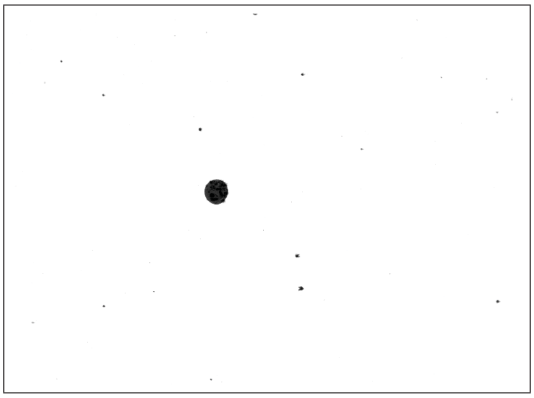


Figure 3.6: An image processed with the function *Imadjust* and the chosen values: Input:0.25;0.7 Output:0;1 gamma=0.8. The result is an image where the inclusion can be detected clearly.

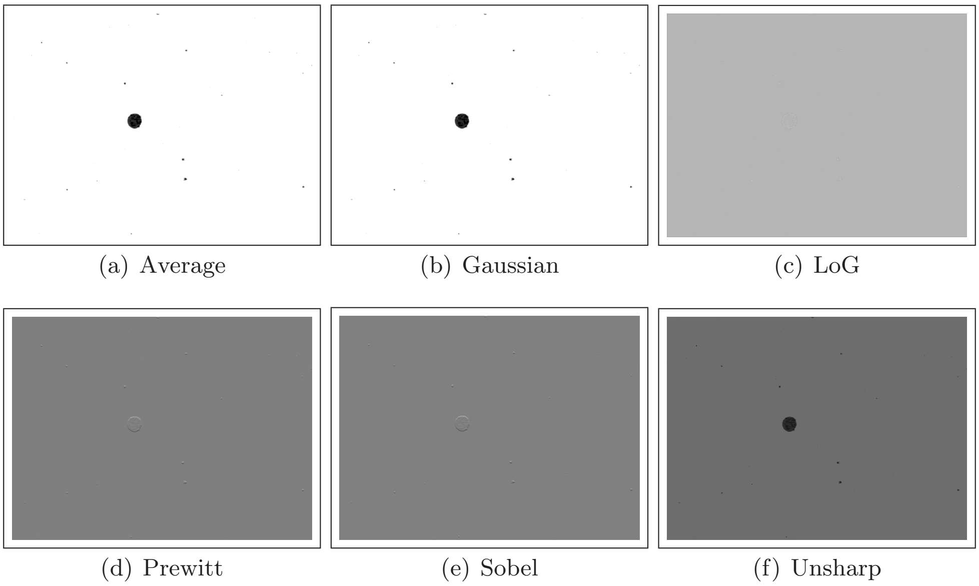
### 3.2.2 Spatial Filtering

To eliminate remaining noise or scratches different filters (see Section 2.1.5) were tested on the pre-processed image (Fig. 3.7).

The *Average Filter* (see Eq. 2.7) and the *Gaussian Filter*(see Eq. 2.9), which are used for smoothing and noise reduction, delivered almost identical results (Fig. 3.7(a) and Fig. 3.7(b)), which could be used for further processing.

The results of the *Laplacian of Gaussian Filter* (Eq. 2.12), *Prewitt Filter* (Eq. 2.15 and 2.16) and the *Sobel Filter* (Eq. 2.21 and 2.22), which are used as masks for *Edge Detection*, delivered results which turned out to be unsuitable for further processing (Fig. 3.7(c),Fig. 3.7(d) and Fig. 3.7(e)). The gray level value difference between inclusions and background was too low to find an

appropriate threshold. *Thresholding* is used for creating a convincing binary image, which is needed for receiving morphological information.
The *Unsharp Filter* (see Eq. 2.23 and Fig. 3.7(f)) delivered results similar to the *Average Filter* and the *Gaussian Filter*. With these filters an appropriate threshold could be defined.
For further processing the *Unsharp Filter* was used, because smaller inclusions were separated best from the background.



(a) Average

(b) Gaussian

(c) LoG

(d) Prewitt

(e) Sobel

(f) Unsharp

Figure 3.7: Different filter functions have been used to eliminate noise and scratches in the images.

### 3.2.3 Thresholding

After *Intensity Transformation* and *Spatial Filtering* the image is not yet a binary image, which is needed for gaining morphological information about the inclusions. Therefore *Thresholding* is used. Two thresholds ($T1, T2$), which represent the intensity boarders of the inclusions, were defined. Oxides and sulfides have different intensities. Therefore two different threshold operations have been used.
To find the right threshold for oxides a test inclusion (Fig. 3.8) with different threshold values was analyzed. As determining value the length of the boundary (in pixels) was chosen. Table 3.2 shows that boundary lengths don't differ much. That means that a variation of threshold values in a small range isn't very sensitive. 112 and 167 were chosen as the threshold values for oxides (see Fig. 3.9). That is to say all values below 112 and above 167 are set to 1 (white) and values between 112 and 167 are set to 0 (Eq. 3.1).
For sulfides the thresholds 155 and 185 were chosen: All values below 155 and above 185 are set to 1 and values between 155 and 185 are set to 0 (Eq. 3.2). That preserves the morphology of the inclusion in the binary image best.

$$g(x,y)_{oxide} = \begin{cases} 0 & if\ 111 < f(x,y) < 168 \\ 1 & if\ f(x,y) < 112\ or f(x,y) > 167 \end{cases} \quad (3.1)$$

$$g(x,y)_{sulfide} = \begin{cases} 0 & if\ 155 < f(x,y) < 185 \\ 1 & if\ f(x,y) < 155\ or f(x,y) > 185 \end{cases} \quad (3.2)$$

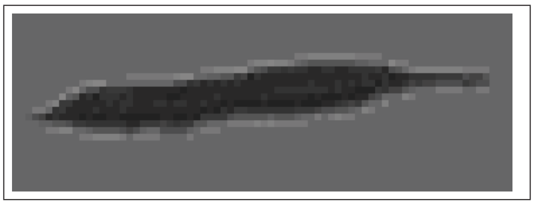


Figure 3.8: Test oxide inclusion to receive adequate threshold values.

Table 3.2: Boundary length values for the test inclusion with different threshold values.

| | **140** | **150** | **160** | **167** | **180** | **190** | **210** |
|---|---|---|---|---|---|---|---|
| **130** | | 300 | 307 | 308 | 312 | 316 | 332 |
| **125** | | 300 | 306 | 307 | 309 | 314 | 329 |
| **120** | 286 | 297 | 304 | 307 | 309 | 314 | 329 |
| **115** | 284 | 294 | 300 | 306 | 309 | 312 | 328 |
| **112** | 284 | 293 | 300 | **306** | 308 | 312 | 324 |
| **105** | | 286 | 297 | 301 | 307 | 309 | 317 |
| **100** | | 284 | 294 | 300 | 307 | 309 | 316 |

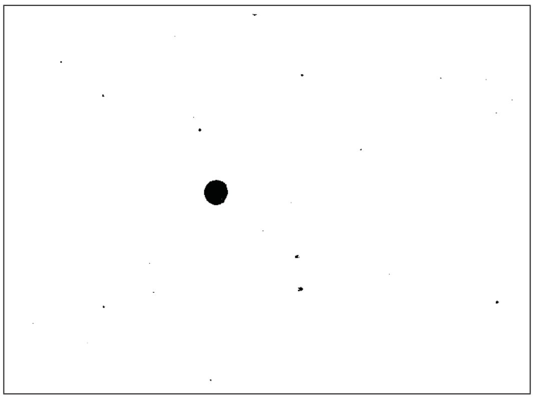
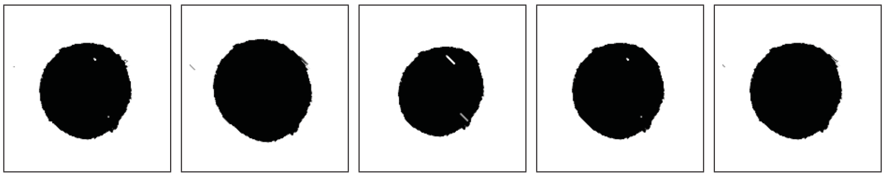


Figure 3.9: Binary image with an oxide type inclusion after *Thresholding* with 112 and 167.

### 3.2.4 Morphological Operators

The image is now arranged with the mathematical operations *Dilation*, *Erosion*, *Opening* and *Closing* to change the image in a way that morphological features like *Area* or *Bounding Box* can be calculated better. Therefore *Erosion* can be used to erode noise and *Dilation* can be used to fill holes in connected regions. Figure 3.10 shows an example of an inclusion where the structure element is an $8 \times 8$ matrix of zeros with ones in diagonal direction.
Figure 3.10(a) shows the original image. Figure 3.10(b) and Fig. 3.10(c) show the results of *Dilation* and *Erosion*. The dilated object grows (Eq. 2.24) and the eroded object shrinks (Eq. 2.25). Figure 3.10(d) and Fig. 3.10(e) show the result of *Opening* and *Closing*. *Opening* is an *Erosion* followed by a *Dilation*. In this operation the holes stay in the object because the image is eroded first. *Closing* is a *Dilation* followed by an *Erosion*. In this operation the holes go missing because the image is dilated first. The operations *Opening* and *Closing* were tested to see if they can deliver better results than *Erosion* and *Dilation*, but for further usage only *Erosion* was used to erode leftover noise.

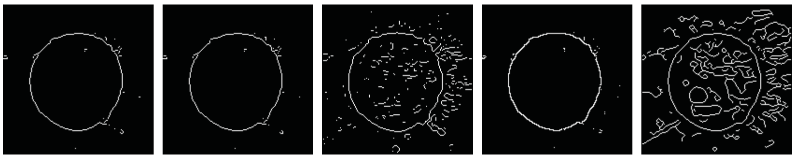


(a) Original (b) *Dilation* - The boundaries are "growing" (c) *Erosion* - The boundaries are "shrinking" (d) *Opening* (e) *Closing*

Figure 3.10: An OG type inclusion processed with various mathematical operations to erode noise and to fill holes in connected regions.

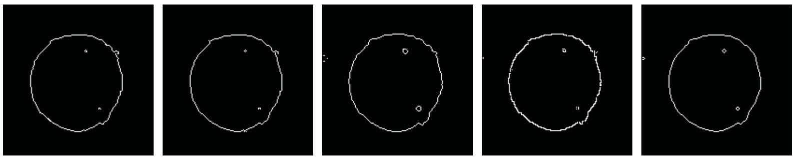
### 3.2.5 Edge Detection

In order to define connected areas, different edge detectors like *Sobel, Prewitt, LoG, Roberts* and *Canny* were tested (see Section 2.1.7). With *Edge Detection* the edges of connected areas are computed in order to apply morphological processing.
Figure 3.11(a), 3.11(b) and 3.11(d) show the *Sobel*, the *Prewitt* and the *Roberts* edge detectors, which delivered results where noise was excluded but the shapes of the boundaries weren't continuous. Figure 3.11(c) and 3.11(e) show the results of *LoG* and the *Canny* edge detectors. Noise and small gray level changes were detected but the boundaries had a connected shape.
With different steps of pre-processing it was able to optimize *Edge Detection*. The appearance of noise was almost deleted (Fig. 3.12). The *Canny* edge detector delivered the best results for continuous edges.



(a) *Sobel* (b) *Prewitt* (c) *LoG* (d) *Roberts* (e) *Canny*

Figure 3.11: An OG type inclusion processed with different types of *Edge Detection* operators. No pre-processing was used.



(a) *Sobel* (b) *Prewitt* (c) *LoG* (d) *Roberts* (e) *Canny*

Figure 3.12: An OG type inclusion processed with different types of *Edge Detection* operators. Pre-processing with methodologies like *Filtering* and *Morphological Operations* was used.

### 3.2.6 Regiongrow

*Regiongrow* is an operation that finds connected pixels or subregions that belong to a larger region [2]. While a region grows from a defined seed point adjacent pixels are tested if they correlate to predefined properties. In this work the neighborhood pixels are tested if they have the same gray level values as the predefined seed points. If they have, they belong to the same region. Seed points can be defined as matrix or scalar and the properties can be defined as a matrix or a scalar too. If a scalar is chosen, it equals a threshold value. The points with intensity value 0 were defined as seed points, threshold was defined to be 0.9, giving rise to a binary image. Figure 3.13 shows the image after *Regiongrow*, as the last step of pre-processing.

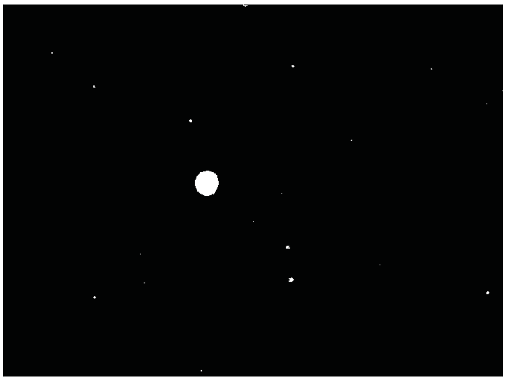


Figure 3.13: Image after *Regiongrow* as last step of pre-processing resulting in a binary image.

## 3.3 Classification of Inclusions

After pre-processing, with the result of a binary image, inclusions in the image can be extracted and classified. Therefore two methodologies are described in this section: The *Morphological Classification* to get the morphological characteristics and the *Classification by Co-occurrence Matrices* to get texture features and type differentiation.

### 3.3.1 Morphological Classification

For *Morphological Classification* a matrix including a wide range of descriptors for morphological characteristics was computed. Therefore the MATLAB function *Regionprops* was used. In this work the descriptors are *Bounding Box*, *Area*, *Centroid*, *Major Axis Length*, *Minor Axis Length* and *Eccentricity.*

Definition of used descriptors:

**Bounding Box:** *"The smallest rectangle containing a region"*. It is defined by 4 values (Fig. 3.14(a) red). The one and two values are the coordinates of the lefthead edgepoint, while value three and four describe the lateral lengths of the rectangle.

**Area:** *"The number of pixels in a region"* (Fig. 3.14(a) white).

**Centroid:** *"The center of mass of a region"* (Fig. 3.14(a) magenta). The values are the coordinates.

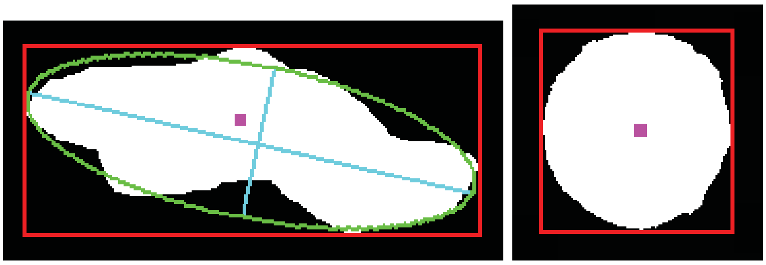
$$x_c = \frac{\sum_i (x_{c,i} \cdot A_i)}{\sum_i A_i};\ y_c = \frac{\sum_i (y_{c,i} \cdot A_i)}{\sum_i A_i} \tag{3.3}$$

**Major Axis Length, Minor Axis Length:** *"The length of the major and minor axis of the ellipse that has the same second moment as the region"* (Fig. 3.14(a) (a) green, blue).
Like in physics Moments are weighted averages. In image processing moments are weighted averages of the pixel intensities. They are used to describe objects after segmentation. Equation 3.4 shows the second moment, where $f(x, y)$ describes an input image [41].

$$M = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} xyf(x, y)dxdy \tag{3.4}$$

**Eccentricity:** *"The eccentricity of the ellipse that has the same second moment as the region"* (Fig. 3.14(a) green). That is to say the ratio of the minor axis and the major axis. If the *Eccentricity* is 1, the region is circle-shaped.

The inclusion in the image (Fig. 3.14(b)) is almost a circle so eccentricity is 1.05 (Tab. 3.3).



(a) A schematic inclusion to describe the used descriptors

(b) OG type inclusion with eccentricity of 1.05

Figure 3.14: Definition of the morphological descriptors computed with *Regionprops*. Bounding Box (red), Area (white), Centroid (magenta), ellipse with same second moment (green), Major and Minor Axes (blue).

Table 3.3: Morphological descriptors for Fig. 3.14(b).

| | |
|---|---|
| **Bounding Box** | 30; 33 99; 104 |
| **Area** | 7988 |
| **Centroid** | 79; 83 |
| **Major Axis Length** | 103.53 |
| **Major Axis Length** | 98.37 |
| **Eccentricity** | 1.05 |

### 3.3.2 Classification by the Utilization of Co-occurrence Matrices

The MATLAB function *Graycomatrix* computes a *Gray Level Co-occurrence Matrix* (GLCM) from an image matrix. The occurrence of two pixels with defined gray levels and a defined distance is stored in these matrices (for the theory see Section 2.2).

Therefore *Graycomatrix* has a few parameters to define:

**GrayLimits:** Defines if gray levels are excluded. In this work the default settings, the minimum and maximum graylevels, were chosen, so every gray level value was stored in the GLCM.

**NumLevels:** Defines the size of the GLCM. That means the gray levels are scaled to downsize the matrix. The matrix size was set to be 10 × 10.

**Offset:** Defines the direction and the distance between two pixels (see Fig. 3.15).

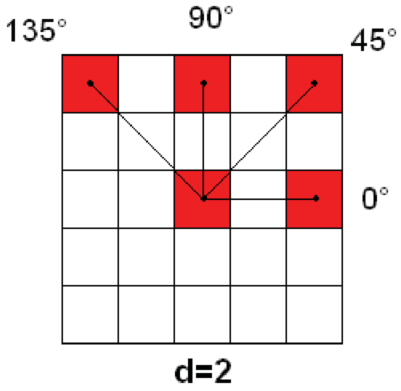


Figure 3.15: Parameters to compute a GLCM: 4 defined directions with distance d=2.

**Symmetric:** A symmetric matrix was chosen. That means the occurrence of pixels was detected in both directions (for example +45° and -45°, see equation 3.5).

The image matrices used in this work include a steel matrix and dark inclusions of varying size. Hence, for creating the GLCMs the inclusions had to be inspected separately. Therefore the original image matrix was multiplied with the image matrix after preprocessing in order to set the background pixels 0. After this operation only the inclusion pixels are left and can be used for creating a GLCM.

For the OG type inclusion from Fig. 3.14(b), which is the biggest inclusion of Fig. 3.2(a), the GLCM in direction 0° and for the distance d=10 is shown in Equation 3.5.

$$GLCM_{0,10} = \begin{bmatrix} 2368 & 0 & 0 & 0 & 117 & 515 & 310 & 225 & 129 & 14 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 117 & 0 & 0 & 0 & 2070 & 2174 & 97 & 54 & 29 & 0 \\ 515 & 0 & 0 & 0 & 2174 & 5236 & 422 & 239 & 109 & 7 \\ 310 & 0 & 0 & 0 & 97 & 422 & 90 & 36 & 24 & 4 \\ 225 & 0 & 0 & 0 & 54 & 239 & 36 & 66 & 20 & 2 \\ 129 & 0 & 0 & 0 & 29 & 109 & 24 & 20 & 10 & 3 \\ 14 & 0 & 0 & 0 & 0 & 7 & 4 & 2 & 3 & 0 \end{bmatrix} \tag{3.5}$$

With the function *Graycoprops* the *Haralick Features* (see Tab. 2.1) can be calculated. In this work the features *Angular Second Moment* (ASM, see Eq. 2.35), *Contrast* (CON, see Eq. 2.36) and *Correlation* (CORR, see Eq. 2.37) were used (see Tab. 3.4).
It turned out that later on the *Haralick Features* have not been used because most inclusions are oxides and have similar characteristics. To differentiate sulfides from oxides the method with Haralick features is too time-consuming. For further investigations, where inclusions should be separated from errors like holes or dust, it is a method that should kept in mind.

Table 3.4: Haralick features for the OG type inclusion from Fig. 3.14(b).

| Angle | ASM | Contrast | Correlation |
|---|---|---|---|
| 0° | 0.1349 | 5.5560 | 0.3687 |
| 45° | 0.1279 | 5.6593 | 0.2142 |
| 90° | 0.1316 | 5.5675 | 0.3710 |
| 135° | 0.1248 | 5.5875 | 0.2242 |
| Avg. | 0.1298 | 5.5926 | 0.2945 |

GLCMs have also been used to define the oxide type of the detected inclusions. The first step was calculating a GLCM for a small distance to find out on which matrix position the highest occurrence value appears. Then GLCMs were computed for four directions and with different distances $d$. The occurrence value on the predefined position was selected and stored.
In Eq. 3.5 the highest occurrence value appears for grayscale category 6 (position 6,6 in the matrix: 5236). The distances 2, 4, 8, 10, 15, 20, 30, 50, 60, 80 and 100 pixels were used and the received value on matrix position 6,6 was outputted. So values for every direction and distance were stored and the result can be used like a ”target” which runs from pixel to pixel. Therefore five values for every distance were used: four values described the occurrence in a defined direction and one value described the sum for all directions.
Figure 3.16 shows how the ”target” was used with a schematic example of a clear situation: The ”target” overlays an inclusion. The origin pixel is equal to the center of the target. If the target coincide with the inclusion at least one target point corresponds to the origin pixel. The corresponding pixel is defined through a direction and a distance. If the values in all four directions aren’t zero, the origin pixel has at least four corresponding pixels. If a higher distance delivers just zeros the inclusion can detected as globular oxide (OG) because the shape of the inclusion is between two circles.
If some values of the four directions are zero and a higher distance delivers just ”close to” results, the inclusion can be detected as line shaped oxide (OS) or dissolved oxides (OA) because the shape is more line shaped than globular.

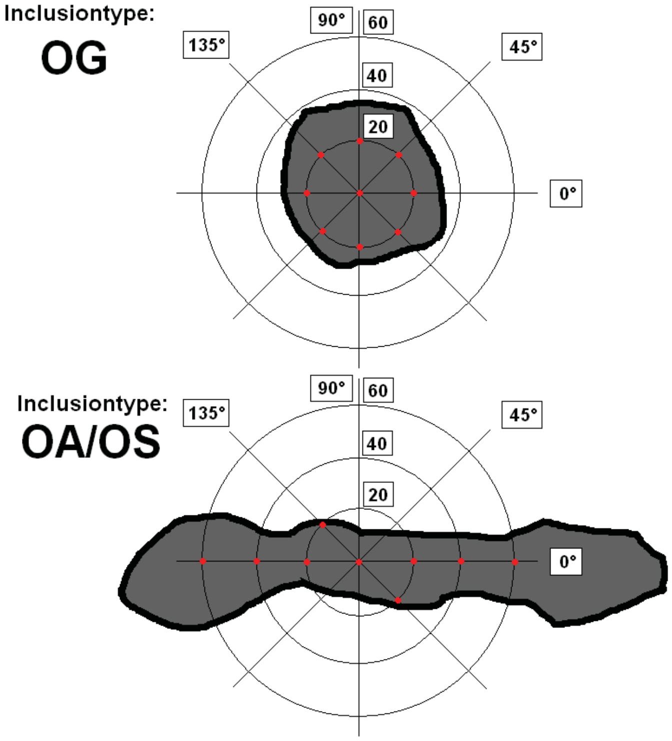


Figure 3.16: A line shaped inclusion (OS) and a globular inclusion (OG) overlayed with the ”target” that runs from pixel to pixel. The red pixels on the ”target” are corresponding to the origin pixel.

# 4 Results of the Classification of Non-Metallic Tool Steel Inclusions

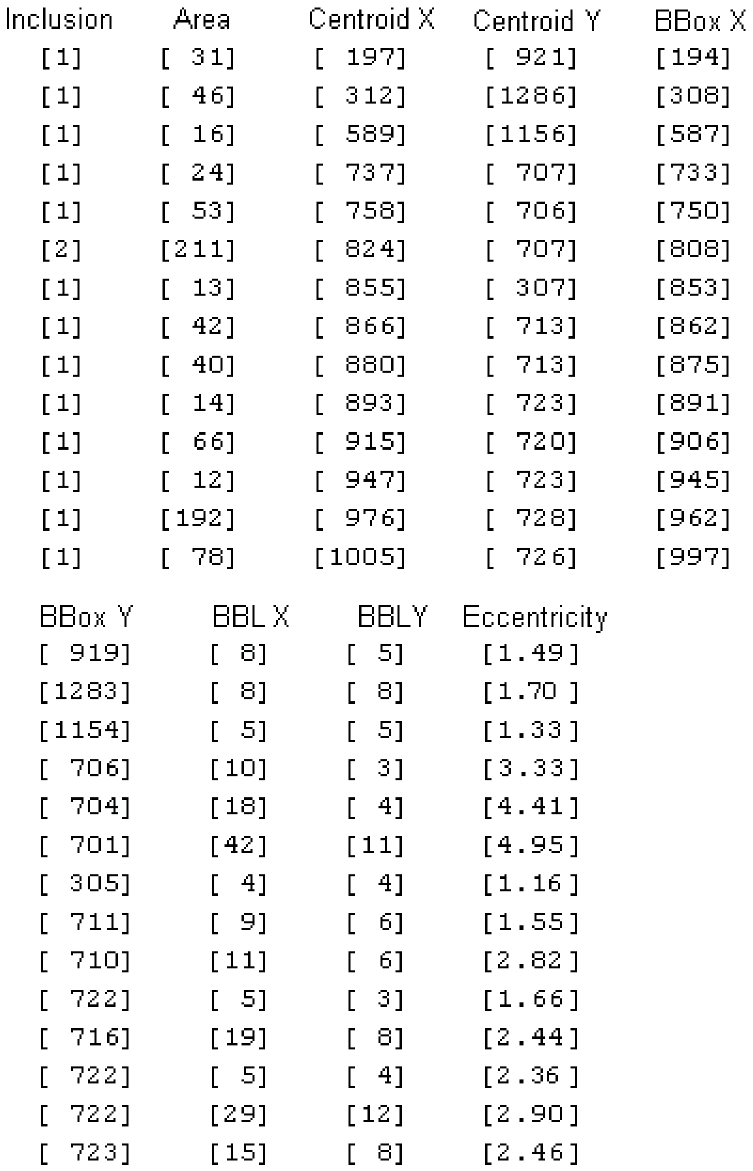
In a specimen a lot of regions are detectable and the aim is to find which region is an inclusion and specifically of what type and which size it is. In this section the classification of oxide and sulfide inclusions is shown including the classification of the shape of the inclusion (line-shaped, dissolved, globular).

## 4.1 Classification of Oxides

Oxide inclusions appear in three types: line-shaped (OS), dissolved (OA) and globular (OG). After pre-processing using a threshold (see Eq. 3.1) to differentiate from sulfide inclusions a lot of calculations had to be done to get the type of the oxide inclusion. These calculations are presented with the aid of five images in the following subsections. Subsection 4.1.1-4.1.3 show an example for every oxide type. Therefore the same processing was used. Subsection 4.1.4 show the refinings of the processing.

### 4.1.1 Classification of Dissolved Oxide Inclusions - OA

An example is discussed in the following and it is shown how the matrices were generated (Fig. 4.1, inclusion type OA). Every region is viewed separately at first and in further operation steps they are combined to a connected inclusion. The matrix $M$ contains the morphological results for every detected region, calculated as shown in Section 3.3.1. The first column of $M$ defines the inclusion type with the ”target” method, where 1 stands for OG and 2 stands for OA/OS. Every region is viewed separately. Thats why nearly all regions are stored as OG in the matrix $M$. In Fig. 4.1 it is shown that the small parts (regions) of the inclusion have a globular shape. Column 2 defines the area, Column 3 and 4 the x- and y-coordinates of the centroid, column 5 and 6 the x- and y-coordinates of the bounding box lefthead, column 7 and 8 the side-lengths of the bounding box and column 9 the eccentricity (see Tab. 4.1). The first inclusion is an OG type with an area of 31 pixels, the centroid is at (197,921), the lefthead edge point of the bounding box is at (194,919), the bounding box has a length of 8 pixels and a width of 5 pixels and the eccentricity is 1.49.
In Matlab regions are counted along the x-axis, that means they are counted from the left to the right, so in some cases the calculated order of regions doesn’t agree with the real order. This is one of the problems in connecting regions. The Figures 4.2 and 4.3 show the way of numbering regions in Matlab schematically. The regions are numbered from the left to right and every inclusion is marked by a green box. Region 3 and 5 are two parts of one inclusion for instance



| Inclusion | Area | Centroid X | Centroid Y | BBox X | BBox Y | BBL X | BBLY | Eccentricity |
|---|---|---|---|---|---|---|---|---|
| [1] | [ 31] | [ 197] | [ 921] | [194] | [ 919] | [ 8] | [ 5] | [1.49] |
| [1] | [ 46] | [ 312] | [1286] | [308] | [1283] | [ 8] | [ 8] | [1.70 ] |
| [1] | [ 16] | [ 589] | [1156] | [587] | [1154] | [ 5] | [ 5] | [1.33] |
| [1] | [ 24] | [ 737] | [ 707] | [733] | [ 706] | [10] | [ 3] | [3.33] |
| [1] | [ 53] | [ 758] | [ 706] | [750] | [ 704] | [18] | [ 4] | [4.41] |
| [2] | [211] | [ 824] | [ 707] | [808] | [ 701] | [42] | [11] | [4.95] |
| [1] | [ 13] | [ 855] | [ 307] | [853] | [ 305] | [ 4] | [ 4] | [1.16] |
| [1] | [ 42] | [ 866] | [ 713] | [862] | [ 711] | [ 9] | [ 6] | [1.55] |
| [1] | [ 40] | [ 880] | [ 713] | [875] | [ 710] | [11] | [ 6] | [2.82] |
| [1] | [ 14] | [ 893] | [ 723] | [891] | [ 722] | [ 5] | [ 3] | [1.66] |
| [1] | [ 66] | [ 915] | [ 720] | [906] | [ 716] | [19] | [ 8] | [2.44] |
| [1] | [ 12] | [ 947] | [ 723] | [945] | [ 722] | [ 5] | [ 4] | [2.36 ] |
| [1] | [192] | [ 976] | [ 728] | [962] | [ 722] | [29] | [12] | [2.90] |
| [1] | [ 78] | [1005] | [ 726] | [997] | [ 723] | [15] | [ 8] | [2.46] |

Table 4.1: Matrix $M$, which contains the inclusion type and the morphological descriptors for image 1 (Fig. 4.1).

Figure 4.1: Image1: Specimen with inclusion type OA.

and region 4 is a separate inclusion, but Matlab orders region 4 between 3 and 5 at the first place. Anyway in later calculations the regions 3, 4 and 5 are recognized as one inclusion.

The second matrix is the distance matrix $D$. In this matrix the distances between a region and all neighbor regions in both directions are stored. Figure 4.3 shows schematically how the distances have been calculated. The chosen center region is region 4. The distances between region 4 and the last (region 1-3) and the next three (region 5-7) regions were measured. Column 1 to 3 of the distance matrix $D$ contain the distances between the computed region and the three left regions, column 4 contains the index, the columns 5 to 7 contains the distances between the computed region and the three regions at the right side and column 8 and 9 are the centroid coordinates (see Tab. 4.2). If regions aren't in a circumcircle of 120 around the computed region and their area is smaller than 100, the index (column 4) is set as 0. In this case the region is detected as an outlier and that means it won't be used for later calculations, because it is a separate inclusion or an error. A separate algorithm checks if it is an inclusion or an error. If it is a discrete inclusion the region is added to the matrix $R$ . Region 4 (index) has a distance of 473 pixels to region 3, 718 pixels to region 2, 581 pixels to region 1, 21 pixels to region 5, 87 pixels to region 6 and 417 pixels to region 7. The centroid is (737,707).

The matrix $D1$ is just the matrix $D$ without the positions where the index was set 0 (see Tab. 4.3).

Matrix $D2$ is a distance matrix that is based on the results of $D1$. In this case the distances between a computed region and the next three regions are measured (see Tab. 4.4). As Example region 1 (complies to region 4 in matrix $D$) has a distance of 21 pixels to region 2, 87 pixels to region 3 and 129 pixels region 4. The centroid is (737,707).

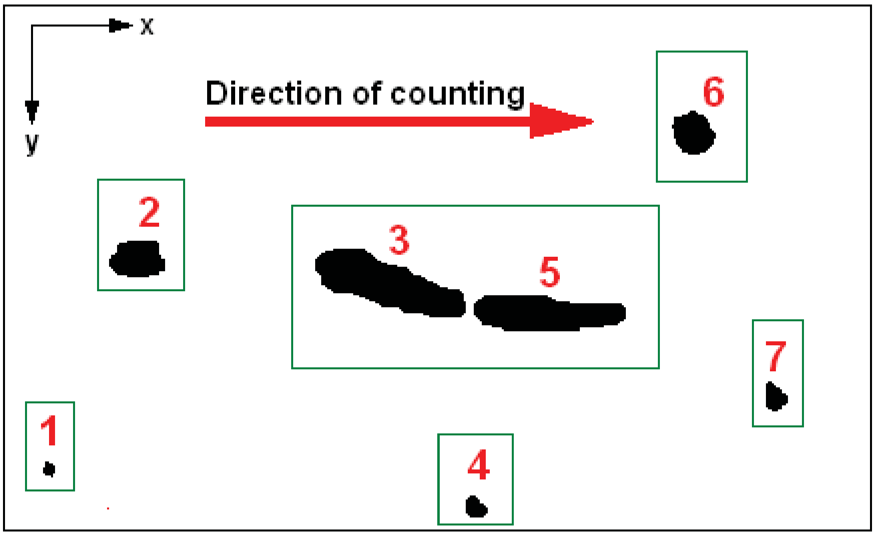


Figure 4.2: The way of numbering regions in Matlab. Regions are numbered from the left to the right, inclusions are marked by a green box. Region 3 and 5 are two parts of one inclusion.

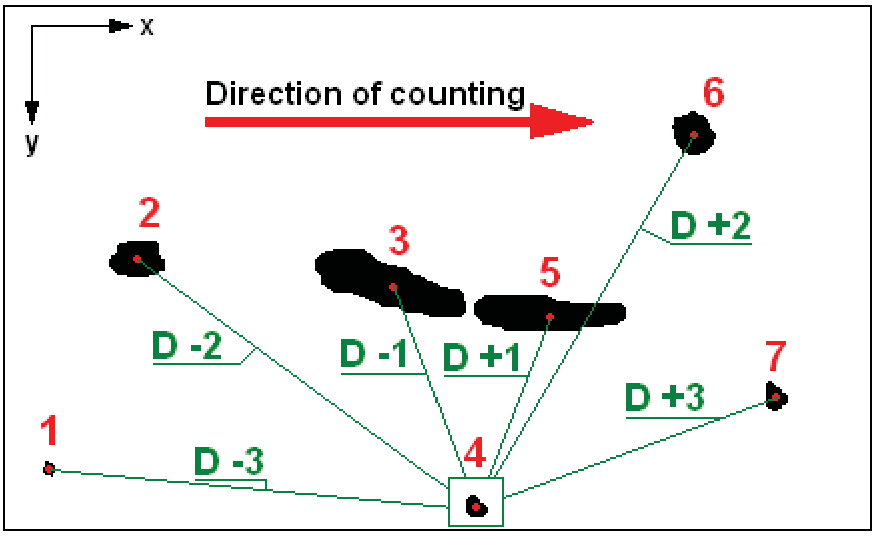
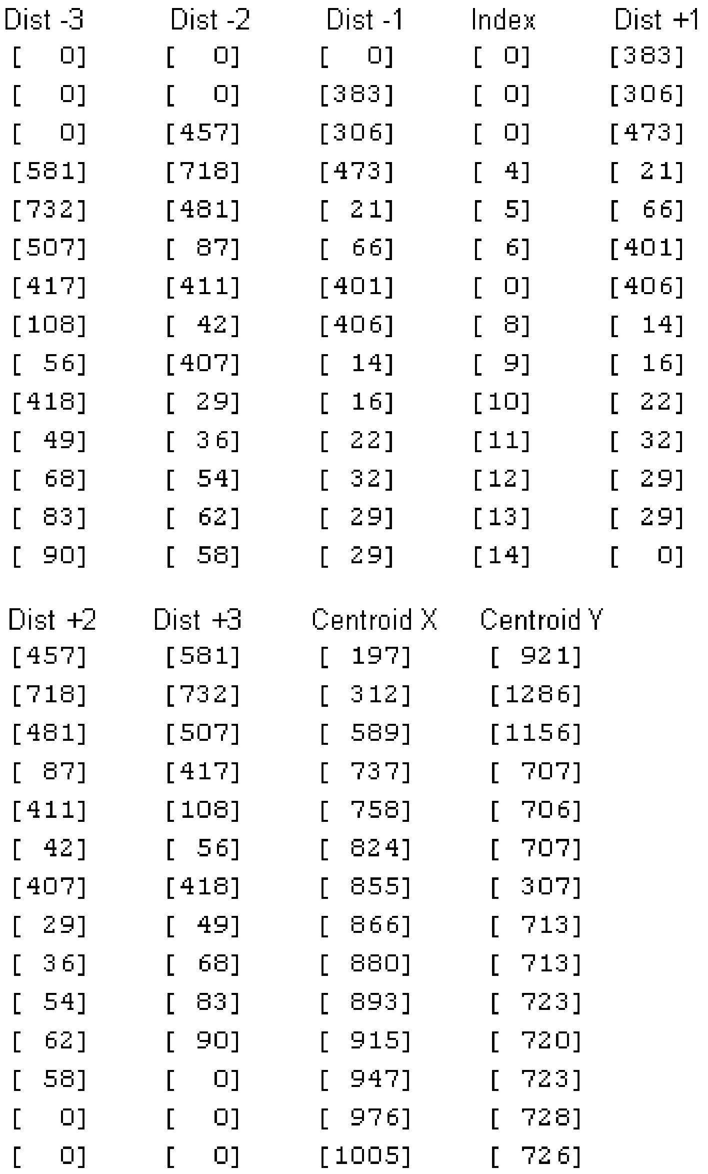


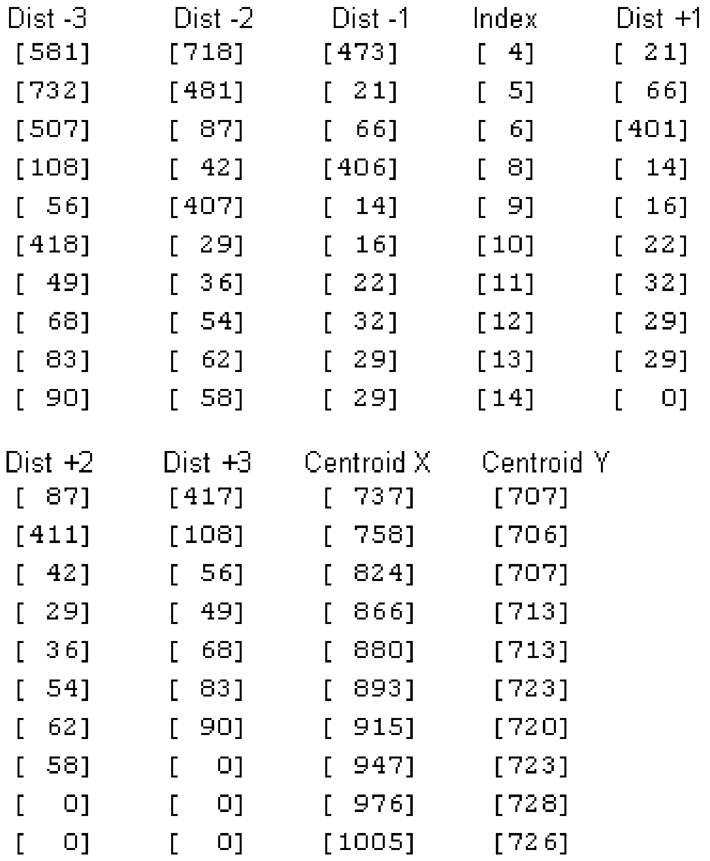
Figure 4.3: Explanation of distance-measurement in the matrix $D$. The distances between region 4 and the other regions are measured.



| Dist -3 | Dist -2 | Dist -1 | Index | Dist +1 |
|---|---|---|---|---|
| [ 0] | [ 0] | [ 0] | [ 0] | [383] |
| [ 0] | [ 0] | [383] | [ 0] | [306] |
| [ 0] | [457] | [306] | [ 0] | [473] |
| [581] | [718] | [473] | [ 4] | [ 21] |
| [732] | [481] | [ 21] | [ 5] | [ 66] |
| [507] | [ 87] | [ 66] | [ 6] | [401] |
| [417] | [411] | [401] | [ 0] | [406] |
| [108] | [ 42] | [406] | [ 8] | [ 14] |
| [ 56] | [407] | [ 14] | [ 9] | [ 16] |
| [418] | [ 29] | [ 16] | [10] | [ 22] |
| [ 49] | [ 36] | [ 22] | [11] | [ 32] |
| [ 68] | [ 54] | [ 32] | [12] | [ 29] |
| [ 83] | [ 62] | [ 29] | [13] | [ 29] |
| [ 90] | [ 58] | [ 29] | [14] | [ 0] |

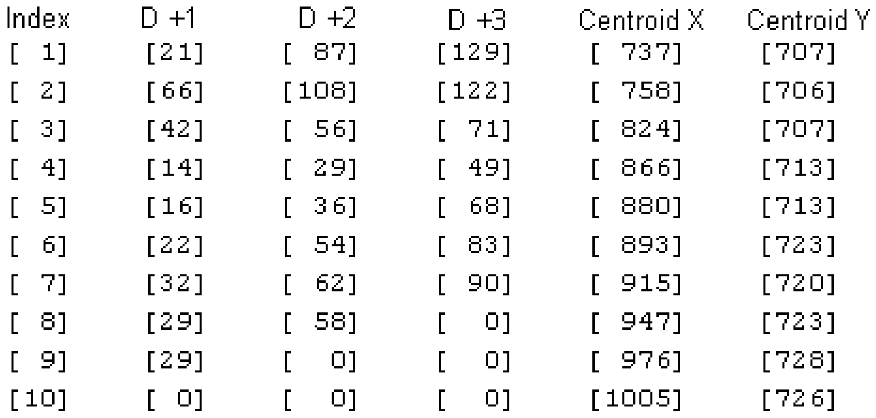
| Dist +2 | Dist +3 | Centroid X | Centroid Y |
|---|---|---|---|
| [457] | [581] | [ 197] | [ 921] |
| [718] | [732] | [ 312] | [1286] |
| [481] | [507] | [ 589] | [1156] |
| [ 87] | [417] | [ 737] | [ 707] |
| [411] | [108] | [ 758] | [ 706] |
| [ 42] | [ 56] | [ 824] | [ 707] |
| [407] | [418] | [ 855] | [ 307] |
| [ 29] | [ 49] | [ 866] | [ 713] |
| [ 36] | [ 68] | [ 880] | [ 713] |
| [ 54] | [ 83] | [ 893] | [ 723] |
| [ 62] | [ 90] | [ 915] | [ 720] |
| [ 58] | [ 0] | [ 947] | [ 723] |
| [ 0] | [ 0] | [ 976] | [ 728] |
| [ 0] | [ 0] | [1005] | [ 726] |

Table 4.2: Distance matrix $D$ with the measured distances for six regions and the coordinates of the centroid for image 1 (Fig. 4.1).



| Dist -3 | Dist -2 | Dist -1 | Index | Dist +1 | Dist +2 | Dist +3 | Centroid X | Centroid Y |
|---|---|---|---|---|---|---|---|---|
| [581] | [718] | [473] | [ 4] | [ 21] | [ 87] | [417] | [ 737] | [707] |
| [732] | [481] | [ 21] | [ 5] | [ 66] | [411] | [108] | [ 758] | [706] |
| [507] | [ 87] | [ 66] | [ 6] | [401] | [ 42] | [ 56] | [ 824] | [707] |
| [108] | [ 42] | [406] | [ 8] | [ 14] | [ 29] | [ 49] | [ 866] | [713] |
| [ 56] | [407] | [ 14] | [ 9] | [ 16] | [ 36] | [ 68] | [ 880] | [713] |
| [418] | [ 29] | [ 16] | [10] | [ 22] | [ 54] | [ 83] | [ 893] | [723] |
| [ 49] | [ 36] | [ 22] | [11] | [ 32] | [ 62] | [ 90] | [ 915] | [720] |
| [ 68] | [ 54] | [ 32] | [12] | [ 29] | [ 58] | [ 0] | [ 947] | [723] |
| [ 83] | [ 62] | [ 29] | [13] | [ 29] | [ 0] | [ 0] | [ 976] | [728] |
| [ 90] | [ 58] | [ 29] | [14] | [ 0] | [ 0] | [ 0] | [1005] | [726] |

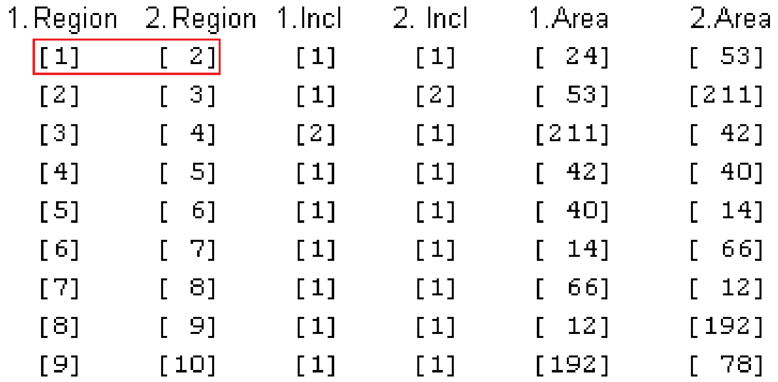
Table 4.3: Distance matrix $D1$ with the measured distances for six regions and the coordinates of the centroid for image 1, without outliers (Fig. 4.1).



| Index | D +1 | D +2 | D +3 | Centroid X | Centroid Y |
|---|---|---|---|---|---|
| [ 1] | [21] | [ 87] | [129] | [ 737] | [707] |
| [ 2] | [66] | [108] | [122] | [ 758] | [706] |
| [ 3] | [42] | [ 56] | [ 71] | [ 824] | [707] |
| [ 4] | [14] | [ 29] | [ 49] | [ 866] | [713] |
| [ 5] | [16] | [ 36] | [ 68] | [ 880] | [713] |
| [ 6] | [22] | [ 54] | [ 83] | [ 893] | [723] |
| [ 7] | [32] | [ 62] | [ 90] | [ 915] | [720] |
| [ 8] | [29] | [ 58] | [ 0] | [ 947] | [723] |
| [ 9] | [29] | [ 0] | [ 0] | [ 976] | [728] |
| [10] | [ 0] | [ 0] | [ 0] | [1005] | [726] |

Table 4.4: Distance matrix $D2$ with the measured distances to the next three regions and the coordinates of the centroid for image 1 (Fig. 4.1).

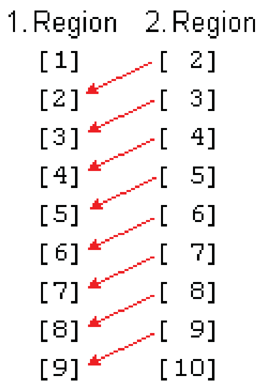
Matrix $D3$ connects the regions stored in $D2$ by pairs. Therefore an algorithm was programmed, where it was checked how the regions are connected. Column 1 stands for the first region, column 2 for the second region, column 3 and 4 show the inclusion types of the region and column 5 and 6 show the areas of the regions (see Tab. 4.5).
For explanation an example where the computed region is region 1 is chosen:
If region D+1 (complies to region 2) is the nearest region and the distance is lower than 130 pixel, the entry in column 1 is *[1]* and in column 2 it is *[2]*: *[1][2]*
If region D+2 (complies to region 3) is the nearest region and the distance is lower than 130 pixel, the entry in column 1 is *[1]* and in column 2 it is *[3]*: *[1][3]*
If region D+3 (complies to region 4) is the nearest region and the distance is lower than 130 pixel, the entry in column 1 is *[1]* and in column 2 it is *[4]*: *[1][4]*
If every three regions have a larger distance than 130, they are sorted out.
In a special case it can be possible that a region is not sorted out, but neither connected with another region. If the computed area of this region is larger than 120 pixels, *[1][1]* is written. If the area is smaller, *[1][0]* is written (see Tab. 4.13).



| 1. Region | 2. Region | 1.Incl | 2. Incl | 1.Area | 2.Area |
|---|---|---|---|---|---|
| [1] | [ 2] | [1] | [1] | [ 24] | [ 53] |
| [2] | [ 3] | [1] | [2] | [ 53] | [211] |
| [3] | [ 4] | [2] | [1] | [211] | [ 42] |
| [4] | [ 5] | [1] | [1] | [ 42] | [ 40] |
| [5] | [ 6] | [1] | [1] | [ 40] | [ 14] |
| [6] | [ 7] | [1] | [1] | [ 14] | [ 66] |
| [7] | [ 8] | [1] | [1] | [ 66] | [ 12] |
| [8] | [ 9] | [1] | [1] | [ 12] | [192] |
| [9] | [10] | [1] | [1] | [192] | [ 78] |

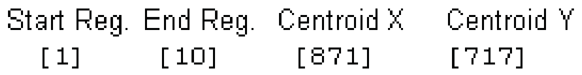
Table 4.5: Distance matrix $D3$ shows if a region of image 1 is connected with the following region. Column 1 and 2 show the connection. *[1][2]* means that region 1 and 2 are connected.

In Matrix $D3$ regions were just connected by pairs, but these regions can belong together too, like it is shown in Tab. 4.6, which shows a part of $D3$. For example two regions with *[1][2]* and *[2][3]* belong to each other, whereas regions with *[1][2]* and *[3][4]* don't belong to each other. Matrix $D4$ connects these regions, where column 1 stands for the start region, column 2 for the end region and column 3 and 4 for the average centroid coordinates (see Tab. 4.7). Region 1 is the start and region 10 the end of the inclusion. The centroid of the whole inclusion is (871,717)
Matrix $D5$ itemizes the different regions of the inclusions. Column 1 stands for the inclusion number, Column 2 for the type of inclusion which was computed in matrix $W$, column 3 is the area and column 4 is the index number (see Tab. 4.8). This matrix serves to define the type of the whole inclusion. For example: If the whole inclusion consists of only one OG, it is an OG type. If there are a



| 1.Region | 2.Region |
|---|---|
| [1] | [ 2] |
| [2] | [ 3] |
| [3] | [ 4] |
| [4] | [ 5] |
| [5] | [ 6] |
| [6] | [ 7] |
| [7] | [ 8] |
| [8] | [ 9] |
| [9] | [10] |

Table 4.6: The first two columns of Distance matrix $D3$ show that all regions belong to the same inclusion (image 1). The end value of the previous region is the start value of the next region.

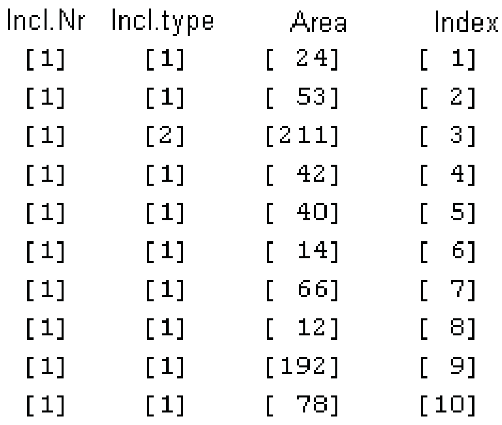


| Start Reg. | End Reg. | Centroid X | Centroid Y |
|---|---|---|---|
| [1] | [10] | [871] | [717] |

Table 4.7: Distance matrix $D4$ shows the start and end region of the inclusion and the average centroid for image 1 (Fig. 4.1).

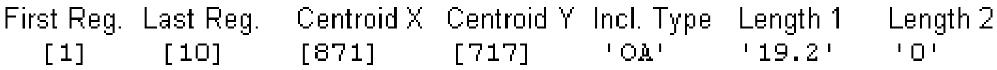
lot of OGs connected to an inclusion, it is an OA type, because it is a dissolved oxide. The result for image 1 is one inclusion with 9 regions of type OG and 1 region of type OA/OS that brings the result of type OA.

The matrix $R$ stores the inclusion types and the length of the inclusions. In the given example the inclusion type is OA and the length is 19.2 $\mu m * 10^{-1}$ , the second length is just used if the type is OG and two small sized OGs are neighbors (see Tab. 4.9). Figure 4.4 shows the image with the inclusion type, the length of the inclusion and the position on the DIN 50 602 board.



| Incl.Nr | Incl.type | Area | Index |
|---|---|---|---|
| [1] | [1] | [ 24] | [ 1] |
| [1] | [1] | [ 53] | [ 2] |
| [1] | [2] | [211] | [ 3] |
| [1] | [1] | [ 42] | [ 4] |
| [1] | [1] | [ 40] | [ 5] |
| [1] | [1] | [ 14] | [ 6] |
| [1] | [1] | [ 66] | [ 7] |
| [1] | [1] | [ 12] | [ 8] |
| [1] | [1] | [192] | [ 9] |
| [1] | [1] | [ 78] | [10] |

Table 4.8: Distance matrix $D5$ shows the different regions of one inclusion for image 1 (Fig. 4.1).



| First Reg. | Last Reg. | Centroid X | Centroid Y | Incl. Type | Length 1 | Length 2 |
|---|---|---|---|---|---|---|
| [1] | [10] | [871] | [717] | 'OA' | '19.2' | '0' |

Table 4.9: Matrix $R$ shows the results for image 1 (Fig. 4.1).

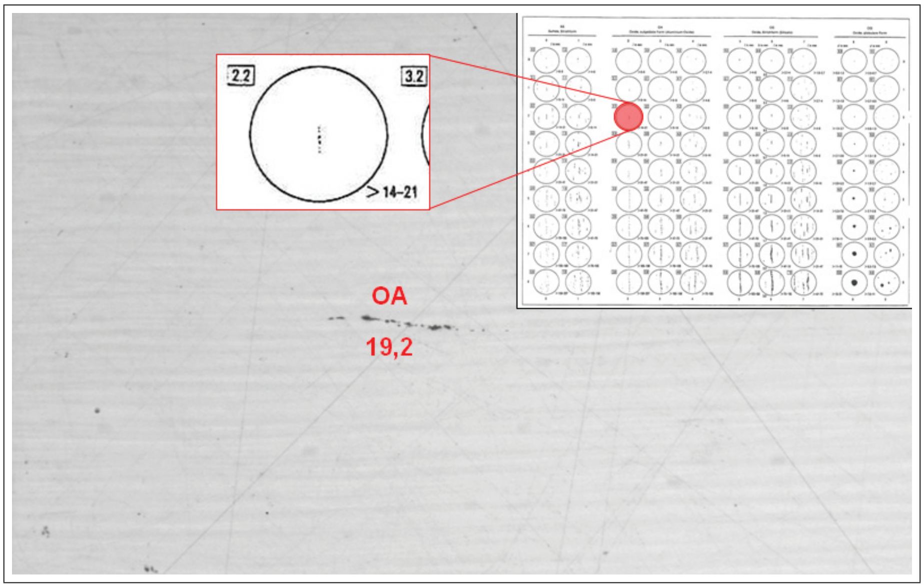


Figure 4.4: Image 1 with the type (OA) and length of the inclusion and its position on the DIN 50 602 board. The crucial length is 19.2 (14<inclusion length<21).

### 4.1.2 Classification of Globular Oxide Inclusions - OG

The second example is an image matrix with inclusions of type OG (see Fig. 4.5). The matrix $M$ detects 19 regions and every region is categorized as 1, meaning OG. The biggest region is region 7 and has an area of 7997 pixels, the centroid at (848,742), the lefthead edgepoint of the bounding box at (799,692), a bounding box with a length of 99 and a width of 104 pixels, and an eccentricity of 1.05 (see Tab. 4.10). Matrix $D$ sorts out 12 regions, so that 7 regions stay in matrix $D1$ (see Tab. 4.11). Matrix $D2$ makes the same operation as matrix $D$ without outliers. For example region 2 (complies to region 7 in matrix $D1$) with the centroid (848,742) has a distance to its next right neighbor of 429 pixels, 534 pixels to the second right neighbor and 1253 pixels to the third neighbor (see Tab. 4.12). Table 4.13 shows that in this case most regions are not connected. For example the biggest region in matrix $D3$ is region 2 with *[2] [2]* in the first two columns, which means it is a single inclusion of type OG. Region 5 is defined as *[5] [0]* - that means this region is too small to be a single inclusion, so it is detected as an error. Region 6 and 7 are connected, but sorted out later because the areas are too small. Table 4.14 shows the leftover regions defined by the start and the end regions. Matrix $D4$ detects four inclusions with the same start and end region, which means it is a single inclusion of type OG. Table 4.15 shows the matrix $R$, which detects four OG type inclusions. Figure 4.6 shows the image with the type, the length and the position on the DIN 50 602 board for these inclusions.

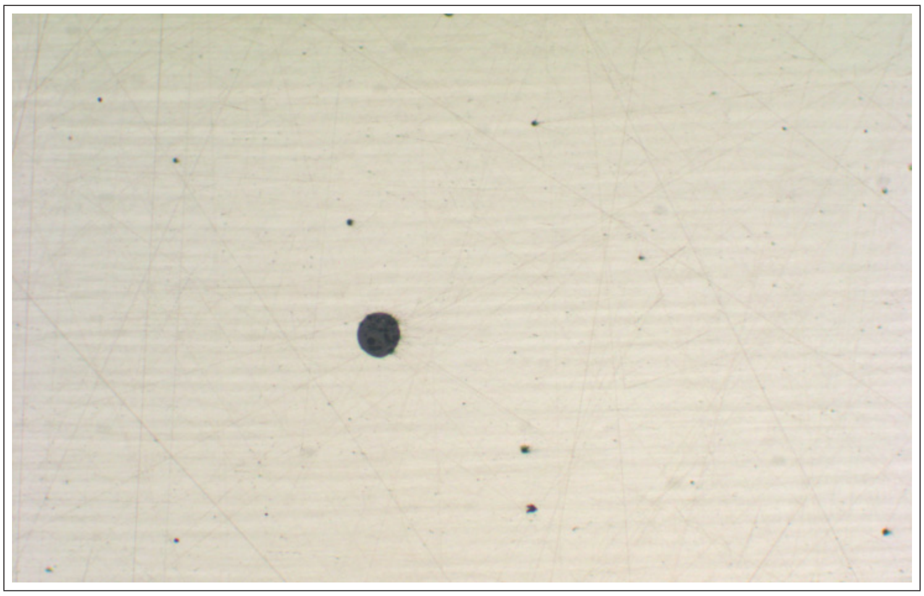
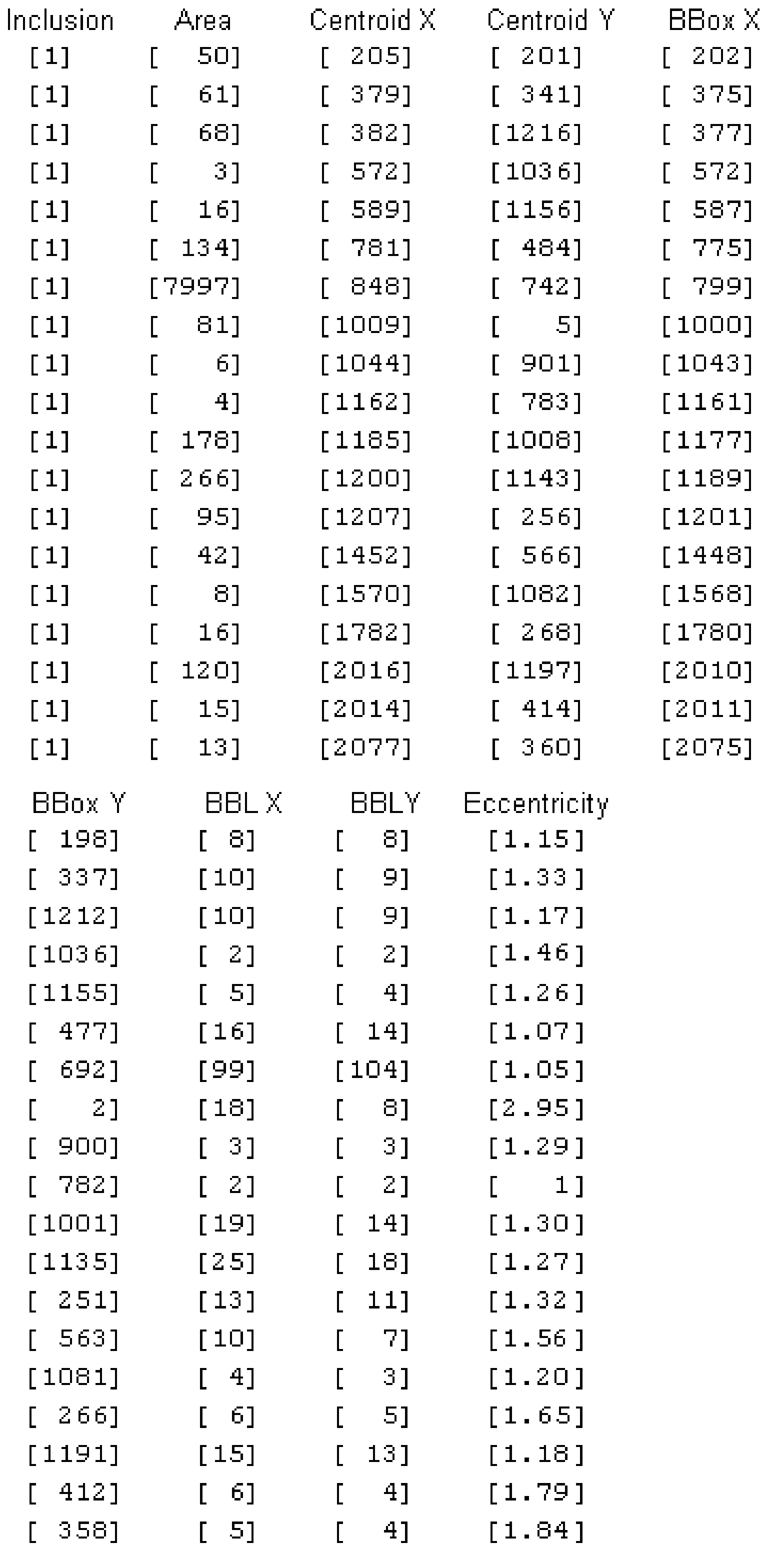
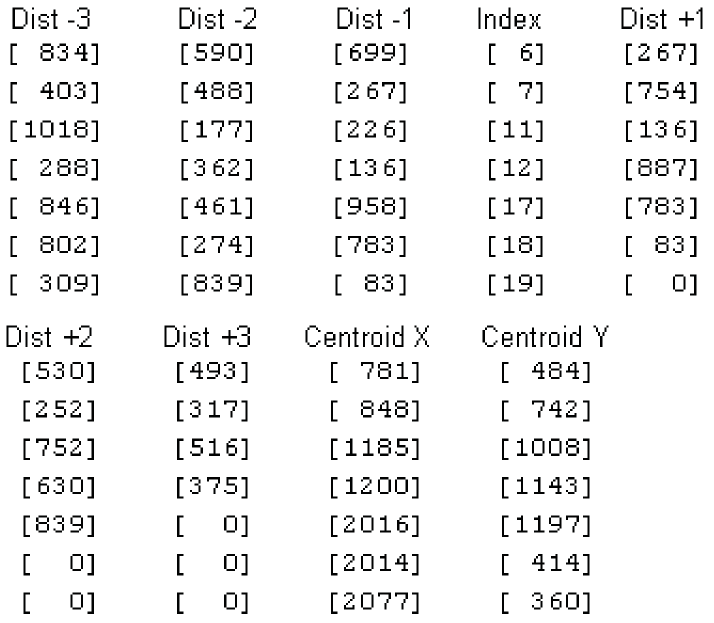


Figure 4.5: Image 2: Specimen with inclusion type OG.



| Inclusion | Area | Centroid X | Centroid Y | BBox X | BBox Y | BBL X | BBLY | Eccentricity |
|---|---|---|---|---|---|---|---|---|
| [1] | [  50] | [ 205] | [ 201] | [ 202] | [ 198] | [ 8] | [  8] | [1.15] |
| [1] | [  61] | [ 379] | [ 341] | [ 375] | [ 337] | [10] | [  9] | [1.33] |
| [1] | [  68] | [ 382] | [1216] | [ 377] | [1212] | [10] | [  9] | [1.17] |
| [1] | [   3] | [ 572] | [1036] | [ 572] | [1036] | [ 2] | [  2] | [1.46] |
| [1] | [  16] | [ 589] | [1156] | [ 587] | [1155] | [ 5] | [  4] | [1.26] |
| [1] | [ 134] | [ 781] | [ 484] | [ 775] | [ 477] | [16] | [ 14] | [1.07] |
| [1] | [7997] | [ 848] | [ 742] | [ 799] | [ 692] | [99] | [104] | [1.05] |
| [1] | [  81] | [1009] | [   5] | [1000] | [   2] | [18] | [  8] | [2.95] |
| [1] | [   6] | [1044] | [ 901] | [1043] | [ 900] | [ 3] | [  3] | [1.29] |
| [1] | [   4] | [1162] | [ 783] | [1161] | [ 782] | [ 2] | [  2] | [   1] |
| [1] | [ 178] | [1185] | [1008] | [1177] | [1001] | [19] | [ 14] | [1.30] |
| [1] | [ 266] | [1200] | [1143] | [1189] | [1135] | [25] | [ 18] | [1.27] |
| [1] | [  95] | [1207] | [ 256] | [1201] | [ 251] | [13] | [ 11] | [1.32] |
| [1] | [  42] | [1452] | [ 566] | [1448] | [ 563] | [10] | [  7] | [1.56] |
| [1] | [   8] | [1570] | [1082] | [1568] | [1081] | [ 4] | [  3] | [1.20] |
| [1] | [  16] | [1782] | [ 268] | [1780] | [ 266] | [ 6] | [  5] | [1.65] |
| [1] | [ 120] | [2016] | [1197] | [2010] | [1191] | [15] | [ 13] | [1.18] |
| [1] | [  15] | [2014] | [ 414] | [2011] | [ 412] | [ 6] | [  4] | [1.79] |
| [1] | [  13] | [2077] | [ 360] | [2075] | [ 358] | [ 5] | [  4] | [1.84] |

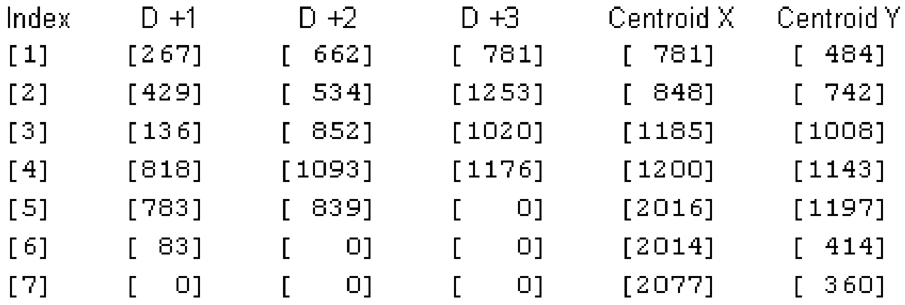
Table 4.10: Matrix $M$, which contains the inclusion type and the morphological descriptors for image 2 (see Fig. 4.5).



| Dist -3 | Dist -2 | Dist -1 | Index | Dist +1 |
|---|---|---|---|---|
| [ 834] | [590] | [699] | [ 6] | [267] |
| [ 403] | [488] | [267] | [ 7] | [754] |
| [1018] | [177] | [226] | [11] | [136] |
| [ 288] | [362] | [136] | [12] | [887] |
| [ 846] | [461] | [958] | [17] | [783] |
| [ 802] | [274] | [783] | [18] | [ 83] |
| [ 309] | [839] | [ 83] | [19] | [ 0] |

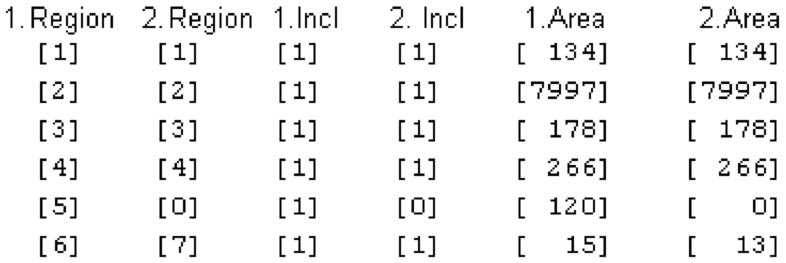
| Dist +2 | Dist +3 | Centroid X | Centroid Y |
|---|---|---|---|
| [530] | [493] | [ 781] | [ 484] |
| [252] | [317] | [ 848] | [ 742] |
| [752] | [516] | [1185] | [1008] |
| [630] | [375] | [1200] | [1143] |
| [839] | [ 0] | [2016] | [1197] |
| [ 0] | [ 0] | [2014] | [ 414] |
| [ 0] | [ 0] | [2077] | [ 360] |

Table 4.11: Distance matrix $D1$ with the measured distances to the least and next three regions and the coordinates of the centroid for image 2, without outliers (see Fig. 4.5).



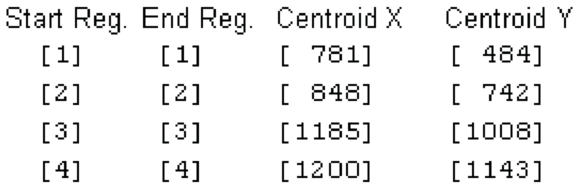
| Index | D +1 | D +2 | D +3 | Centroid X | Centroid Y |
|---|---|---|---|---|---|
| [1] | [267] | [ 662] | [ 781] | [ 781] | [ 484] |
| [2] | [429] | [ 534] | [1253] | [ 848] | [ 742] |
| [3] | [136] | [ 852] | [1020] | [1185] | [1008] |
| [4] | [818] | [1093] | [1176] | [1200] | [1143] |
| [5] | [783] | [ 839] | [ 0] | [2016] | [1197] |
| [6] | [ 83] | [ 0] | [ 0] | [2014] | [ 414] |
| [7] | [ 0] | [ 0] | [ 0] | [2077] | [ 360] |

Table 4.12: Distance matrix $D2$ with the measured distances to three regions and the coordinates of the centroid for image 2 (see Fig. 4.5).



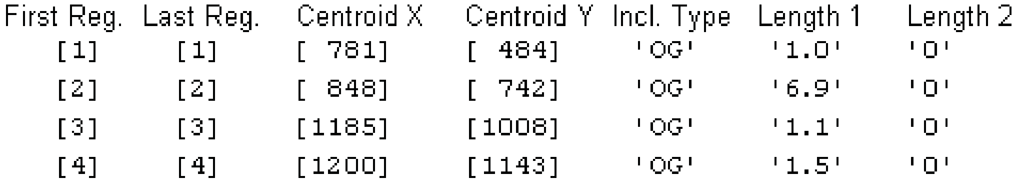
| 1.Region | 2.Region | 1.Incl | 2. Incl | 1.Area | 2.Area |
|---|---|---|---|---|---|
| [1] | [1] | [1] | [1] | [ 134] | [ 134] |
| [2] | [2] | [1] | [1] | [7997] | [7997] |
| [3] | [3] | [1] | [1] | [ 178] | [ 178] |
| [4] | [4] | [1] | [1] | [ 266] | [ 266] |
| [5] | [0] | [1] | [0] | [ 120] | [ 0] |
| [6] | [7] | [1] | [1] | [ 15] | [ 13] |

Table 4.13: Distance matrix $D3$ shows if a region of image 2 is connected with the following region. In this case the regions aren't connected. *[2] [2]* describes a single inclusion, *[5] [0]* a region that is too small for analysis.



| Start Reg. | End Reg. | Centroid X | Centroid Y |
|---|---|---|---|
| [1] | [1] | [ 781] | [ 484] |
| [2] | [2] | [ 848] | [ 742] |
| [3] | [3] | [1185] | [1008] |
| [4] | [4] | [1200] | [1143] |

Table 4.14: Distance matrix $D4$ shows the start and end region of the inclusions and the average centroid for image 2 (see Fig. 4.5).



| First Reg. | Last Reg. | Centroid X | Centroid Y | Incl. Type | Length 1 | Length 2 |
|---|---|---|---|---|---|---|
| [1] | [1] | [ 781] | [ 484] | 'OG' | '1.0' | '0' |
| [2] | [2] | [ 848] | [ 742] | 'OG' | '6.9' | '0' |
| [3] | [3] | [1185] | [1008] | 'OG' | '1.1' | '0' |
| [4] | [4] | [1200] | [1143] | 'OG' | '1.5' | '0' |

Table 4.15: Matrix $R$ shows the results for image 2 (see Fig. 4.5).

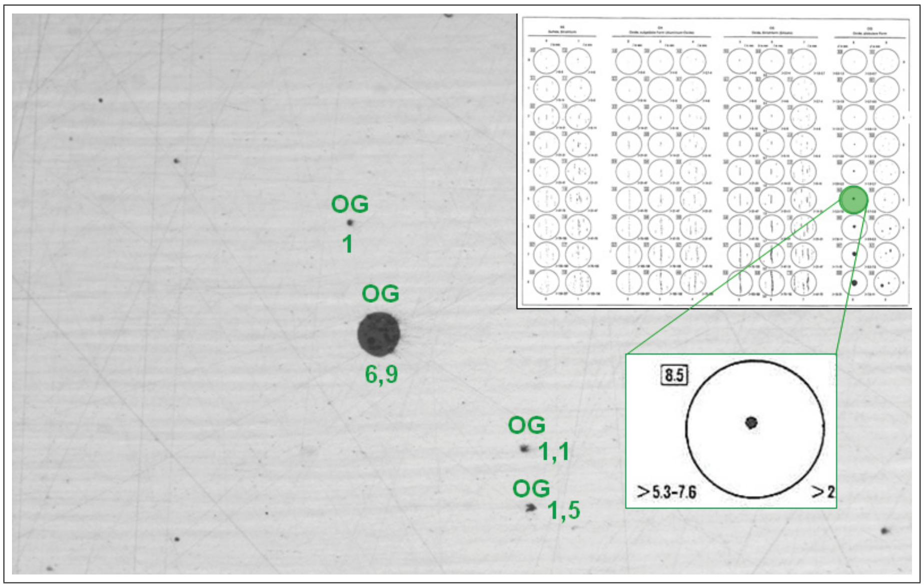


Figure 4.6: Image 2 with type (OG) and length of the inclusion and its position on the DIN 50 602 board. The crucial length is 6.9 (5.3<inclusion length<7.6).

### 4.1.3 Classification of Line-shaped Oxide Inclusions - OS

Figure 4.7 shows an image with three different inclusion types: OA, OG and OS. The image is analyzed with the operations that were tested before (Subsection 4.1.1 and 4.1.2). The matrix $R$ and Fig. 4.7 show the type and length of all detected inclusions in image 3.

| First Reg. | Last Reg. | Centroid X | Centroid Y | Incl. Type | Length 1 | Length 2 |
|---|---|---|---|---|---|---|
| [1] | [4] | [ 867] | [779] | 'OA' | '10.3' | '0' |
| [5] | [6] | [1782] | [526] | 'OS' | ' 6.2' | '0' |
| [5] | [5] | [ 984] | [553] | 'OG' | ' 3.4' | '0' |
| [7] | [7] | [1339] | [533] | 'OS' | ' 6.4' | '0' |

Table 4.16: Matrix $R$ shows the results for image 3 (Fig. 4.7).

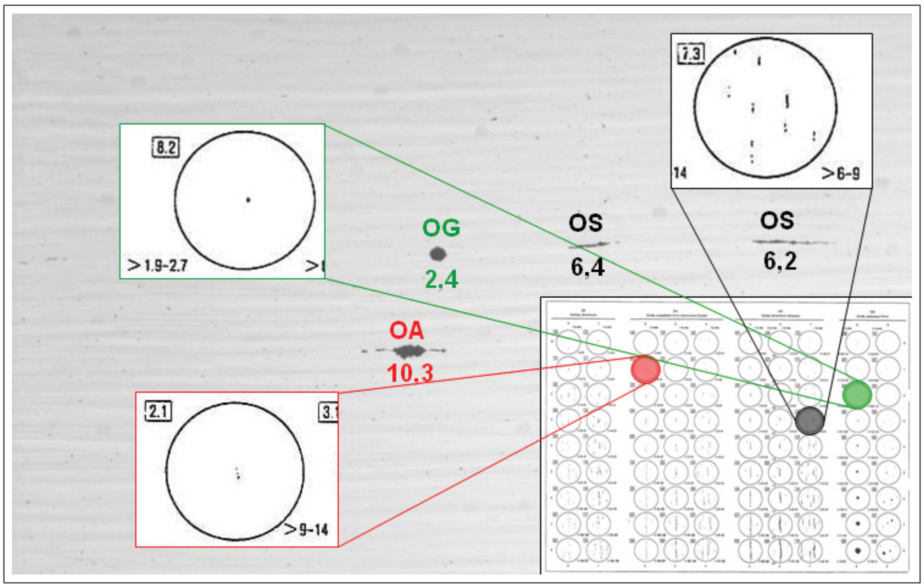


Figure 4.7: Image 3 with type (OA, OG, OS) and length of the inclusions and their positions on the DIN 50 602 board. The crucial lengths are: 10.3 (OA), 2.4 (OG), 6.4 (OS).

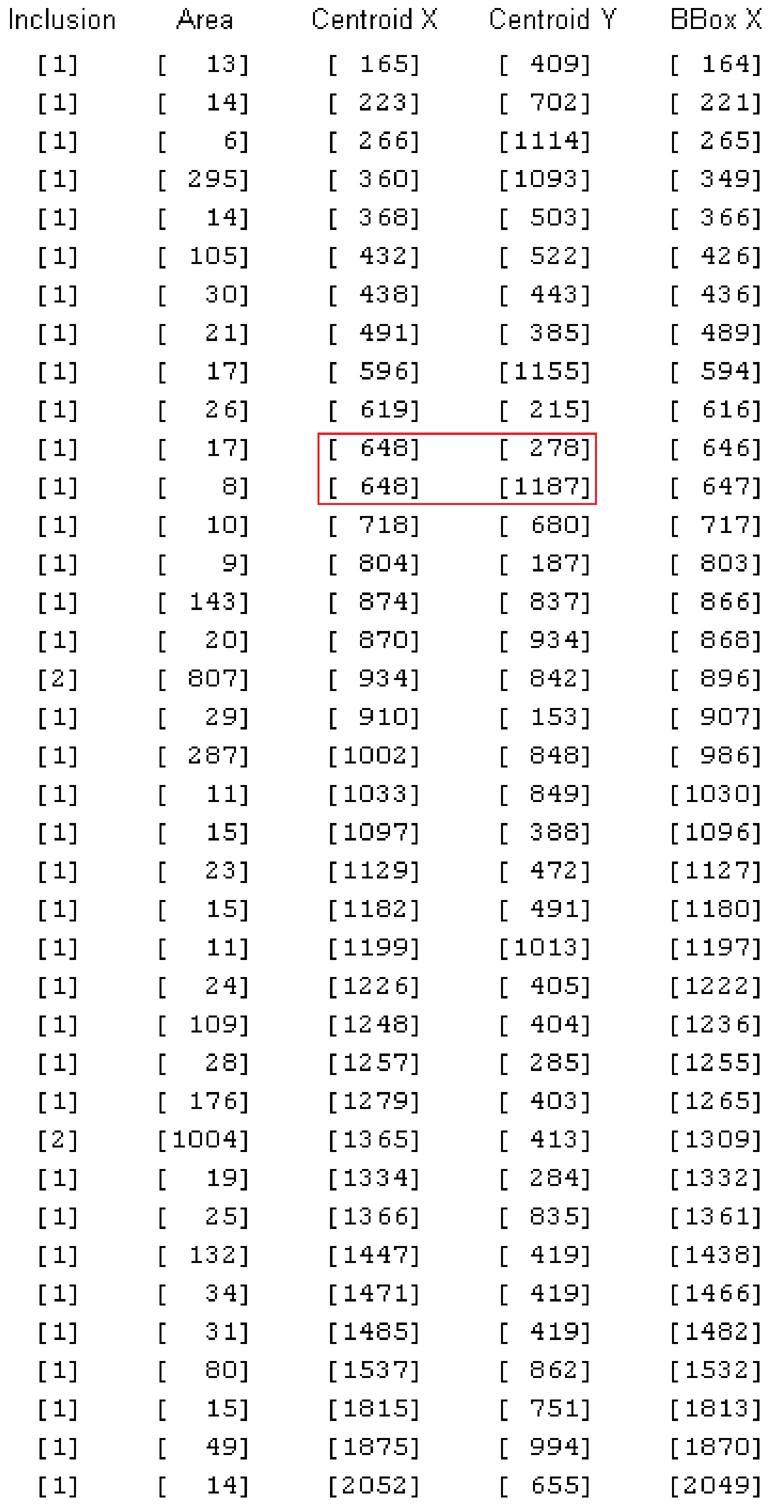
### 4.1.4 Algorithm Improvements for Oxide Classification

Image 4 has inclusions of type OA and OG and shows the problem that is described in Section 4.1.1, Fig. 4.2: Table 4.17 shows that Region 11 and region 12 has the same x-coordinates for the centroid ((648,278) and (648,1187)) and this is a problem for the counting in Matlab.

Figure 4.8: Image 4: Specimen with inclusion types OA and OG.

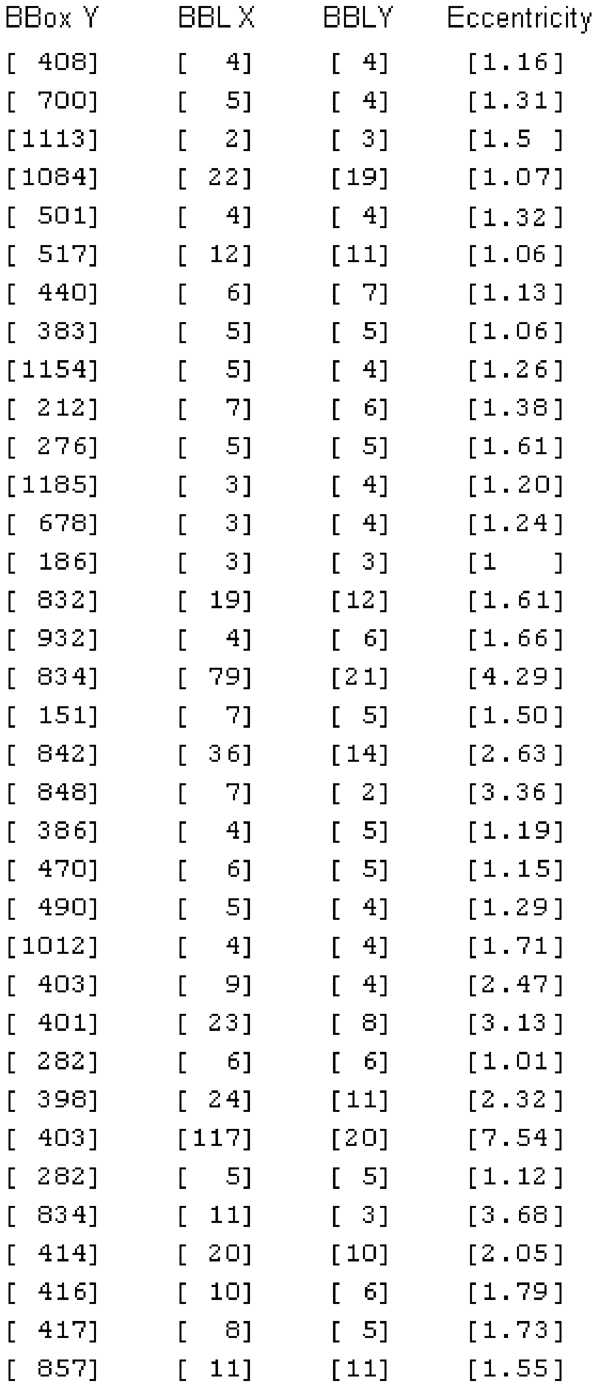
This is a special case because after the calculations to receive the distance matrices, region 10 (complies to region 12 in matrix $M$) isn't detected as an outlier (see Tab. 4.19) and Tab. 4.20 shows that the following regions are connected with this region.

The result of wrong connecting are totally wrong morphological characteristics: The length of inclusion 4 (marked by a red box in Tab. 4.21) was measured from the outlier to the end of the proper inclusion (see Fig. 4.9). To avoid results where various outliers are connected with regions, two submatrices have been programmed and ordered between Matrix $D3$ and $D4$. Submatrix $D2A$ improves the matrix $D2$ and deletes points that are detected as outliers or single inclusions (for example *[1][1]* or *[1][0]* in column 1 and 2). Then the distances between a region and the next three right regions are calculated, like the original matrix $D2$ does (see Tab. 4.22). The deleted points are stored and tested if they are outliers or single inclusions of type OG. If they are detected as inclusions they are later added to the the Matrix $R$. Matrix $D3A$ has the same purpose as matrix $D3$ but is based on matrix $D2A$ for the calculation (see Tab. 4.23). Table 4.24 shows the matrix $D4$ resulting the new start and end regions of inclusions after processing with the submatrices.



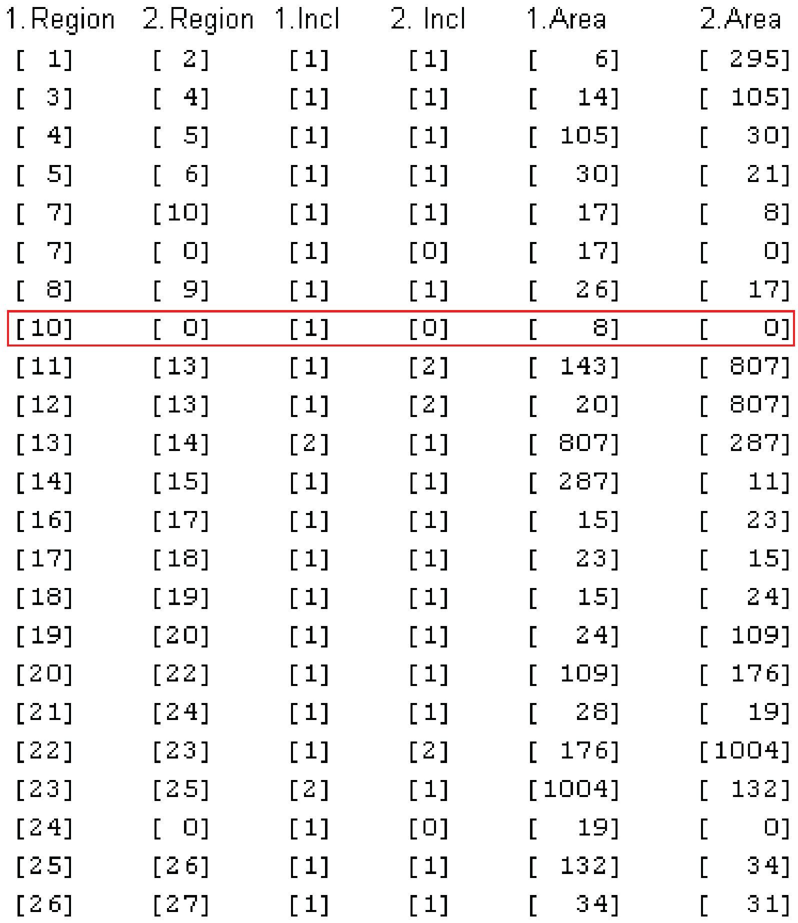
| Inclusion | Area | Centroid X | Centroid Y | BBox X |
|---|---|---|---|---|
| [1] | [ 13] | [ 165] | [ 409] | [ 164] |
| [1] | [ 14] | [ 223] | [ 702] | [ 221] |
| [1] | [ 6] | [ 266] | [1114] | [ 265] |
| [1] | [ 295] | [ 360] | [1093] | [ 349] |
| [1] | [ 14] | [ 368] | [ 503] | [ 366] |
| [1] | [ 105] | [ 432] | [ 522] | [ 426] |
| [1] | [ 30] | [ 438] | [ 443] | [ 436] |
| [1] | [ 21] | [ 491] | [ 385] | [ 489] |
| [1] | [ 17] | [ 596] | [1155] | [ 594] |
| [1] | [ 26] | [ 619] | [ 215] | [ 616] |
| [1] | [ 17] | [ 648] | [ 278] | [ 646] |
| [1] | [ 8] | [ 648] | [1187] | [ 647] |
| [1] | [ 10] | [ 718] | [ 680] | [ 717] |
| [1] | [ 9] | [ 804] | [ 187] | [ 803] |
| [1] | [ 143] | [ 874] | [ 837] | [ 866] |
| [1] | [ 20] | [ 870] | [ 934] | [ 868] |
| [2] | [ 807] | [ 934] | [ 842] | [ 896] |
| [1] | [ 29] | [ 910] | [ 153] | [ 907] |
| [1] | [ 287] | [1002] | [ 848] | [ 986] |
| [1] | [ 11] | [1033] | [ 849] | [1030] |
| [1] | [ 15] | [1097] | [ 388] | [1096] |
| [1] | [ 23] | [1129] | [ 472] | [1127] |
| [1] | [ 15] | [1182] | [ 491] | [1180] |
| [1] | [ 11] | [1199] | [1013] | [1197] |
| [1] | [ 24] | [1226] | [ 405] | [1222] |
| [1] | [ 109] | [1248] | [ 404] | [1236] |
| [1] | [ 28] | [1257] | [ 285] | [1255] |
| [1] | [ 176] | [1279] | [ 403] | [1265] |
| [2] | [1004] | [1365] | [ 413] | [1309] |
| [1] | [ 19] | [1334] | [ 284] | [1332] |
| [1] | [ 25] | [1366] | [ 835] | [1361] |
| [1] | [ 132] | [1447] | [ 419] | [1438] |
| [1] | [ 34] | [1471] | [ 419] | [1466] |
| [1] | [ 31] | [1485] | [ 419] | [1482] |
| [1] | [ 80] | [1537] | [ 862] | [1532] |
| [1] | [ 15] | [1815] | [ 751] | [1813] |
| [1] | [ 49] | [1875] | [ 994] | [1870] |
| [1] | [ 14] | [2052] | [ 655] | [2049] |

Table 4.17: Matrix $M$, which contains the inclusion type and the morphological descriptors for image 4, columns 1 to 5, (Fig. 4.8). The red box marks the regions with the same x-coordinates.



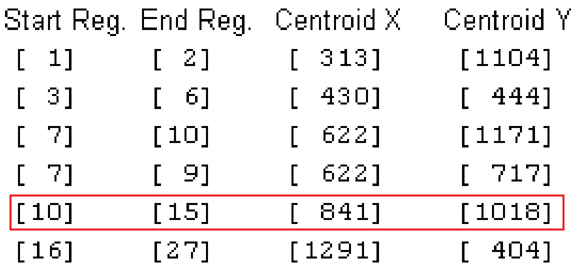
| BBox Y | BBL X | BBLY | Eccentricity |
|---|---|---|---|
| [ 408] | [ 4] | [ 4] | [1.16] |
| [ 700] | [ 5] | [ 4] | [1.31] |
| [1113] | [ 2] | [ 3] | [1.5 ] |
| [1084] | [ 22] | [19] | [1.07] |
| [ 501] | [ 4] | [ 4] | [1.32] |
| [ 517] | [ 12] | [11] | [1.06] |
| [ 440] | [ 6] | [ 7] | [1.13] |
| [ 383] | [ 5] | [ 5] | [1.06] |
| [1154] | [ 5] | [ 4] | [1.26] |
| [ 212] | [ 7] | [ 6] | [1.38] |
| [ 276] | [ 5] | [ 5] | [1.61] |
| [1185] | [ 3] | [ 4] | [1.20] |
| [ 678] | [ 3] | [ 4] | [1.24] |
| [ 186] | [ 3] | [ 3] | [1 ] |
| [ 832] | [ 19] | [12] | [1.61] |
| [ 932] | [ 4] | [ 6] | [1.66] |
| [ 834] | [ 79] | [21] | [4.29] |
| [ 151] | [ 7] | [ 5] | [1.50] |
| [ 842] | [ 36] | [14] | [2.63] |
| [ 848] | [ 7] | [ 2] | [3.36] |
| [ 386] | [ 4] | [ 5] | [1.19] |
| [ 470] | [ 6] | [ 5] | [1.15] |
| [ 490] | [ 5] | [ 4] | [1.29] |
| [1012] | [ 4] | [ 4] | [1.71] |
| [ 403] | [ 9] | [ 4] | [2.47] |
| [ 401] | [ 23] | [ 8] | [3.13] |
| [ 282] | [ 6] | [ 6] | [1.01] |
| [ 398] | [ 24] | [11] | [2.32] |
| [ 403] | [117] | [20] | [7.54] |
| [ 282] | [ 5] | [ 5] | [1.12] |
| [ 834] | [ 11] | [ 3] | [3.68] |
| [ 414] | [ 20] | [10] | [2.05] |
| [ 416] | [ 10] | [ 6] | [1.79] |
| [ 417] | [ 8] | [ 5] | [1.73] |
| [ 857] | [ 11] | [11] | [1.55] |

Table 4.18: Matrix $M$, which contains the inclusion type and the morphological descriptors for image 4, columns 6 to 9 (Fig. 4.8).



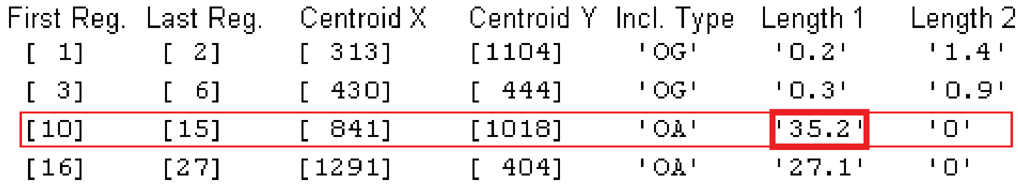
| 1.Region | 2.Region | 1.Incl | 2. Incl | 1.Area | 2.Area |
|---|---|---|---|---|---|
| [ 1] | [ 2] | [1] | [1] | [ 6] | [ 295] |
| [ 3] | [ 4] | [1] | [1] | [ 14] | [ 105] |
| [ 4] | [ 5] | [1] | [1] | [ 105] | [ 30] |
| [ 5] | [ 6] | [1] | [1] | [ 30] | [ 21] |
| [ 7] | [10] | [1] | [1] | [ 17] | [ 8] |
| [ 7] | [ 0] | [1] | [0] | [ 17] | [ 0] |
| [ 8] | [ 9] | [1] | [1] | [ 26] | [ 17] |
| [10] | [ 0] | [1] | [0] | [ 8] | [ 0] |
| [11] | [13] | [1] | [2] | [ 143] | [ 807] |
| [12] | [13] | [1] | [2] | [ 20] | [ 807] |
| [13] | [14] | [2] | [1] | [ 807] | [ 287] |
| [14] | [15] | [1] | [1] | [ 287] | [ 11] |
| [16] | [17] | [1] | [1] | [ 15] | [ 23] |
| [17] | [18] | [1] | [1] | [ 23] | [ 15] |
| [18] | [19] | [1] | [1] | [ 15] | [ 24] |
| [19] | [20] | [1] | [1] | [ 24] | [ 109] |
| [20] | [22] | [1] | [1] | [ 109] | [ 176] |
| [21] | [24] | [1] | [1] | [ 28] | [ 19] |
| [22] | [23] | [1] | [2] | [ 176] | [1004] |
| [23] | [25] | [2] | [1] | [1004] | [ 132] |
| [24] | [ 0] | [1] | [0] | [ 19] | [ 0] |
| [25] | [26] | [1] | [1] | [ 132] | [ 34] |
| [26] | [27] | [1] | [1] | [ 34] | [ 31] |

Table 4.19: Distance matrix $D3$ shows if a region of image 4 is connected with the following region. The red box shows the region that isn't detected as outlier.



| Start Reg. | End Reg. | Centroid X | Centroid Y |
|---|---|---|---|
| [ 1] | [ 2] | [ 313] | [1104] |
| [ 3] | [ 6] | [ 430] | [ 444] |
| [ 7] | [10] | [ 622] | [1171] |
| [ 7] | [ 9] | [ 622] | [ 717] |
| [10] | [15] | [ 841] | [1018] |
| [16] | [27] | [1291] | [ 404] |

Table 4.20: Distance matrix $D4$ shows the start and end region of the inclusions and the average centroid for image 4 (Fig. 4.8). The red box shows the wrong connected inclusion.



| First Reg. | Last Reg. | Centroid X | Centroid Y | Incl. Type | Length 1 | Length 2 |
|---|---|---|---|---|---|---|
| [ 1] | [ 2] | [ 313] | [1104] | 'OG' | '0.2' | '1.4' |
| [ 3] | [ 6] | [ 430] | [ 444] | 'OG' | '0.3' | '0.9' |
| [10] | [15] | [ 841] | [1018] | 'OA' | '35.2' | '0' |
| [16] | [27] | [1291] | [ 404] | 'OA' | '27.1' | '0' |

Table 4.21: Matrix $R$ shows the results for image 4 (Fig. 4.8). The red box shows the wrong connected inclusion. The bold red box shows the wrong length.

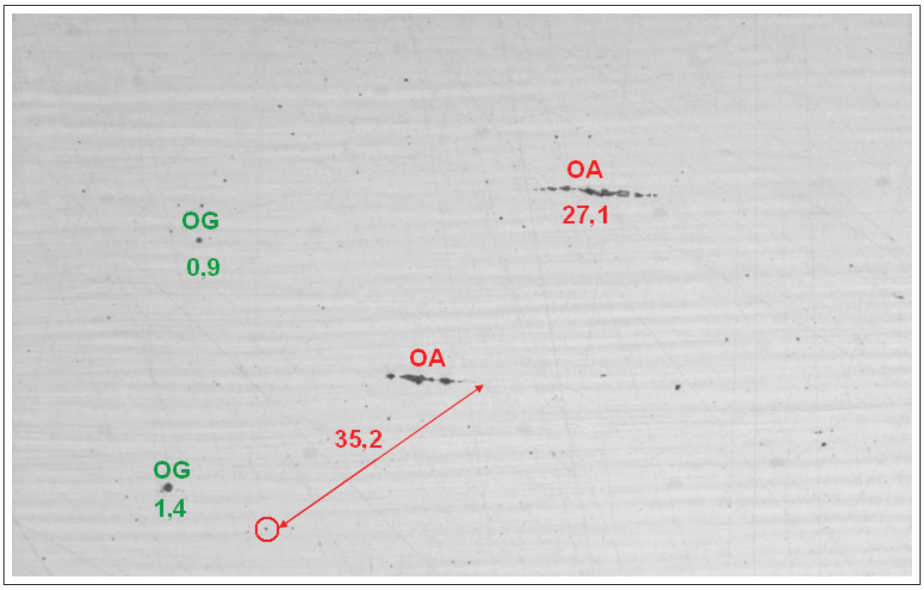
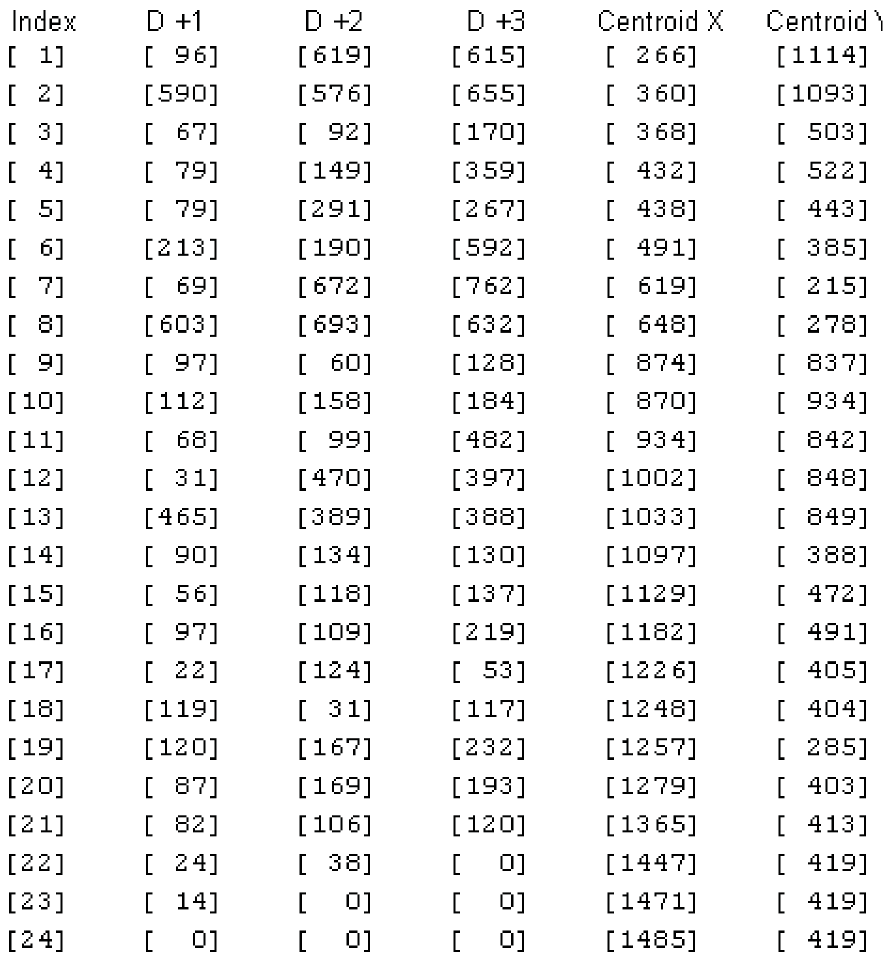
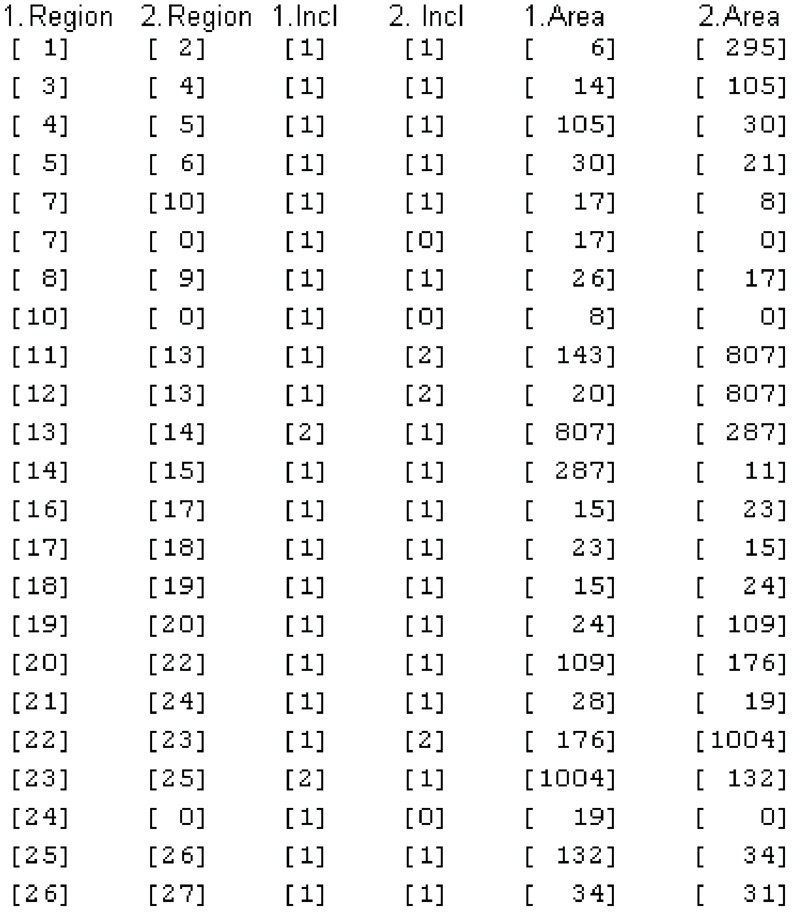


Figure 4.9: Image 4 with the type and length of the inclusions. The arrow marks the length of the wrong connected inclusion.



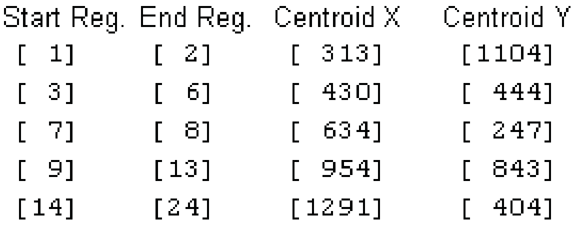
```
Index    D +1     D +2     D +3     Centroid X   Centroid Y
[ 1]     [ 96]    [619]    [615]    [ 266]       [1114]
[ 2]     [590]    [576]    [655]    [ 360]       [1093]
[ 3]     [ 67]    [ 92]    [170]    [ 368]       [ 503]
[ 4]     [ 79]    [149]    [359]    [ 432]       [ 522]
[ 5]     [ 79]    [291]    [267]    [ 438]       [ 443]
[ 6]     [213]    [190]    [592]    [ 491]       [ 385]
[ 7]     [ 69]    [672]    [762]    [ 619]       [ 215]
[ 8]     [603]    [693]    [632]    [ 648]       [ 278]
[ 9]     [ 97]    [ 60]    [128]    [ 874]       [ 837]
[10]     [112]    [158]    [184]    [ 870]       [ 934]
[11]     [ 68]    [ 99]    [482]    [ 934]       [ 842]
[12]     [ 31]    [470]    [397]    [1002]       [ 848]
[13]     [465]    [389]    [388]    [1033]       [ 849]
[14]     [ 90]    [134]    [130]    [1097]       [ 388]
[15]     [ 56]    [118]    [137]    [1129]       [ 472]
[16]     [ 97]    [109]    [219]    [1182]       [ 491]
[17]     [ 22]    [124]    [ 53]    [1226]       [ 405]
[18]     [119]    [ 31]    [117]    [1248]       [ 404]
[19]     [120]    [167]    [232]    [1257]       [ 285]
[20]     [ 87]    [169]    [193]    [1279]       [ 403]
[21]     [ 82]    [106]    [120]    [1365]       [ 413]
[22]     [ 24]    [ 38]    [   0]   [1447]       [ 419]
[23]     [ 14]    [   0]   [   0]   [1471]       [ 419]
[24]     [   0]   [   0]   [   0]   [1485]       [ 419]
```

Table 4.22: Distance matrix $D2A$ with measured distances to the next three regions and coordinates of the centroid for image 4 (Fig. 4.8). $D2A$ is based on $D3$ without outliers.



| 1.Region | 2.Region | 1.Incl | 2. Incl | 1.Area | 2.Area |
|---|---|---|---|---|---|
| [ 1] | [ 2] | [1] | [1] | [ 6] | [ 295] |
| [ 3] | [ 4] | [1] | [1] | [ 14] | [ 105] |
| [ 4] | [ 5] | [1] | [1] | [ 105] | [ 30] |
| [ 5] | [ 6] | [1] | [1] | [ 30] | [ 21] |
| [ 7] | [10] | [1] | [1] | [ 17] | [ 8] |
| [ 7] | [ 0] | [1] | [0] | [ 17] | [ 0] |
| [ 8] | [ 9] | [1] | [1] | [ 26] | [ 17] |
| [10] | [ 0] | [1] | [0] | [ 8] | [ 0] |
| [11] | [13] | [1] | [2] | [ 143] | [ 807] |
| [12] | [13] | [1] | [2] | [ 20] | [ 807] |
| [13] | [14] | [2] | [1] | [ 807] | [ 287] |
| [14] | [15] | [1] | [1] | [ 287] | [ 11] |
| [16] | [17] | [1] | [1] | [ 15] | [ 23] |
| [17] | [18] | [1] | [1] | [ 23] | [ 15] |
| [18] | [19] | [1] | [1] | [ 15] | [ 24] |
| [19] | [20] | [1] | [1] | [ 24] | [ 109] |
| [20] | [22] | [1] | [1] | [ 109] | [ 176] |
| [21] | [24] | [1] | [1] | [ 28] | [ 19] |
| [22] | [23] | [1] | [2] | [ 176] | [1004] |
| [23] | [25] | [2] | [1] | [1004] | [ 132] |
| [24] | [ 0] | [1] | [0] | [ 19] | [ 0] |
| [25] | [26] | [1] | [1] | [ 132] | [ 34] |
| [26] | [27] | [1] | [1] | [ 34] | [ 31] |

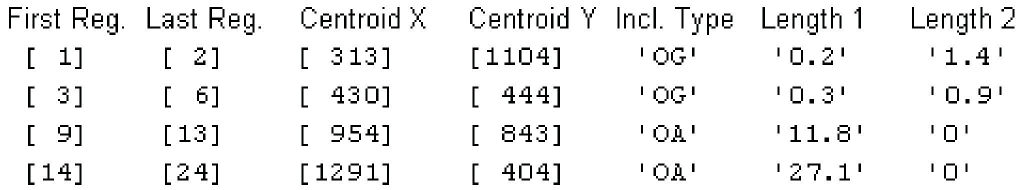
Table 4.23: Distance matrix $D3A$ shows if a region of image 4 is connected with the following region after implementing the submatrices. *[1][2]* means that region 1 and 2 are connected.



| Start Reg. | End Reg. | Centroid X | Centroid Y |
|---|---|---|---|
| [ 1] | [ 2] | [ 313] | [1104] |
| [ 3] | [ 6] | [ 430] | [ 444] |
| [ 7] | [ 8] | [ 634] | [ 247] |
| [ 9] | [13] | [ 954] | [ 843] |
| [14] | [24] | [1291] | [ 404] |

Table 4.24: Distance matrix $D4$ shows the new start and end region of the inclusions and the average centroid for image 4 after implementing the subregions.

After the implementation of submatrices the results are accurate (shown in Tab. 4.25 and Fig. 4.10). Finally for inclusion 3 the correct length of 11.8 $\mu m * 10^{-1}$ instead of 35.2 $\mu m * 10^{-1}$ was detected.



```
First Reg.  Last Reg.   Centroid X   Centroid Y  Incl. Type  Length 1   Length 2
 [ 1]        [ 2]        [ 313]       [1104]       'OG'       '0.2'      '1.4'
 [ 3]        [ 6]        [ 430]       [ 444]       'OG'       '0.3'      '0.9'
 [ 9]        [13]        [ 954]       [ 843]       'OA'       '11.8'     '0'
 [14]        [24]        [1291]       [ 404]       'OA'       '27.1'     '0'
```

Table 4.25: Matrix $R$ shows the results for image 4 (Fig. 4.8).

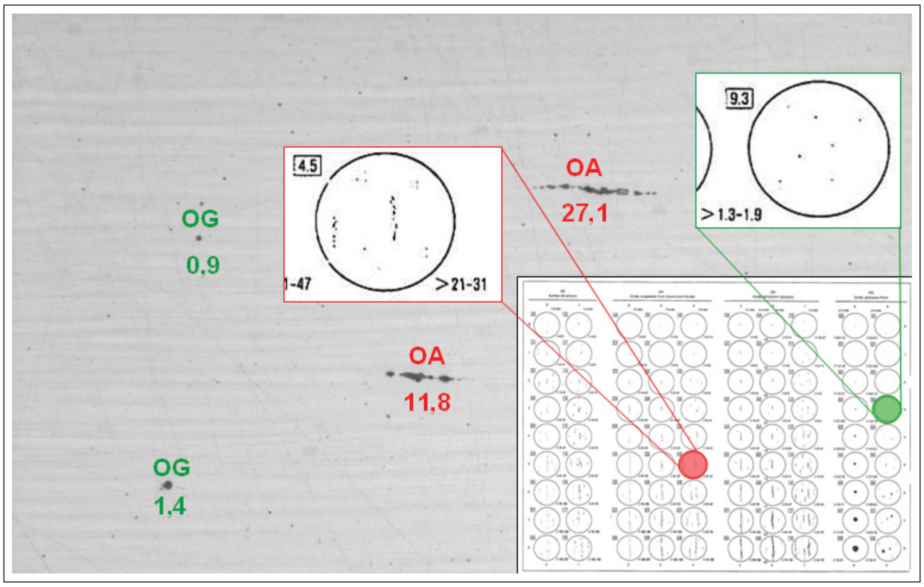


Figure 4.10: Image 4 with type (OA, OG) and length of the inclusions and their positions on the DIN 50 602 board. The crucial lengths are: 27.1 (OA), 1.4 (OG).

The last operation tries to connect leftover regions, if they have a special position that makes a connection possible. Of the whole set of data provided by Böhler Edelstahl GmbH and Co KG and Co KG Fig. 4.11 was the only image where this operation was necessary to use. It is useful when inclusions are big (size index 5 and bigger) and a lot of regions must be connected.
Therefore the distance matrix $D6$ checks if adjacent regions can get connected (see Tab. 4.26). For that purpose the angles between the first and the last region of an inclusion and the angles between adjacent regions have to be measured. To connect adjacent regions the angle between the regions must be in a tolerance field of +/- 15°. The two calculated region angles must fulfill this tolerance too and the distance must be smaller than 200 pixels (see Fig. 4.12). Table 4.26 shows the number of the start region (column 1), the number of the end region (column 2), the inclusion centroid coordinates (column 3 and 4), the centroid

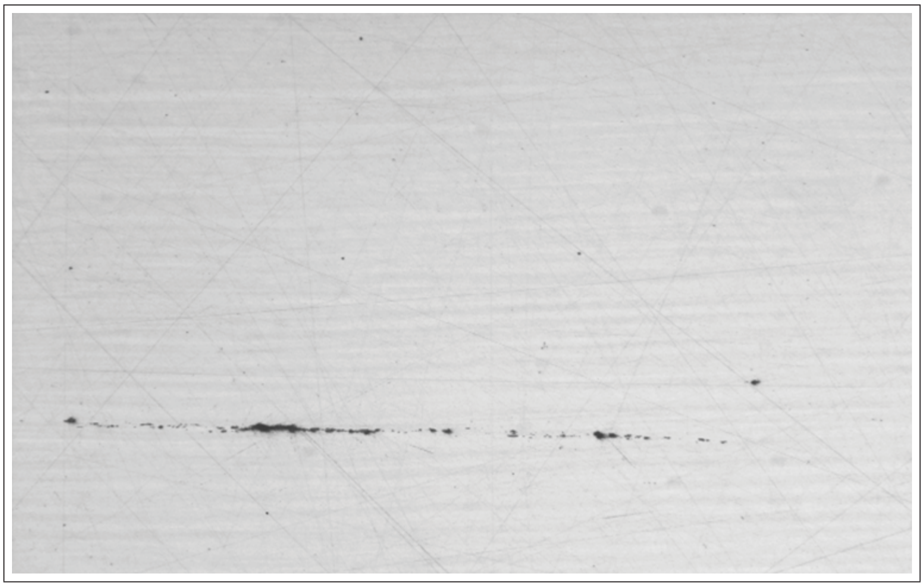


Figure 4.11: Image 5: Specimen with inclusion types OG and OA.

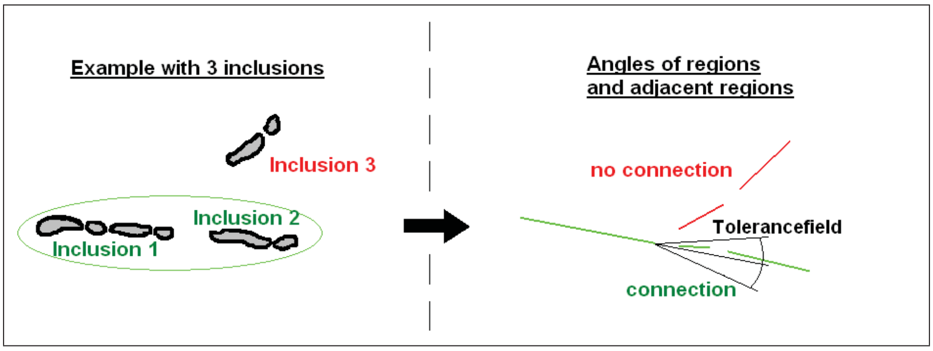
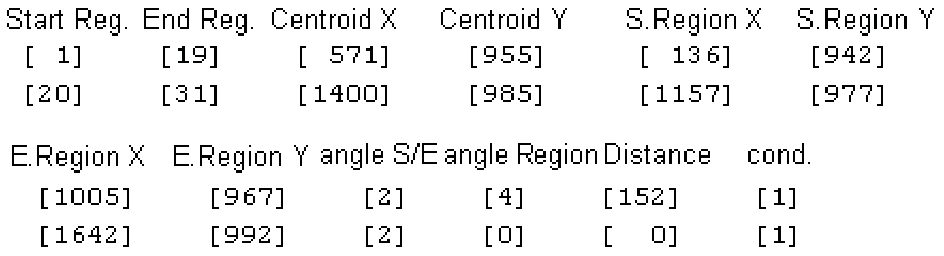


Figure 4.12: Tolerancefield for adjacent regions to connect them.



| Start Reg. | End Reg. | Centroid X | Centroid Y | S.Region X | S.Region Y | E.Region X | E.Region Y | angle S/E | angle Region | Distance | cond. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [ 1] | [19] | [ 571] | [955] | [ 136] | [942] | [1005] | [967] | [2] | [4] | [152] | [1] |
| [20] | [31] | [1400] | [985] | [1157] | [977] | [1642] | [992] | [2] | [0] | [ 0] | [1] |

Table 4.26: Distance matrix *D*6 with the angles between adjacent regions(image 5). Column 11 shows if regions can get connected. 1 means connection and 0 no connection.

coordinates of the start and end region (column 5-8), the angle between the start and the end region (column 9), the angle between two adjacent regions (column 10) and the distance between two adjacent regions (column 10). Column 11 shows if regions can get connected. 1 means connection and 0 no connection.
In this example there are two regions from 1 to 19 and from 20 to 31. The regions have the same angle of 2° (based to the x-axis). The angle between them is 4°. The distance is 152 pixels. The ones in the last column show that those regions must be connected (see Tab. 4.26).
The Matrix $R$ (see Tab. 4.27) and Fig. 4.13 show the results for image 5. The described operation connected two inclusions to one inclusion with the start region 1 and the end region 31. The remaining region is a single inclusion of type OG, which was detected in a processing step before.

| First Reg. | Last Reg. | Centroid X | Centroid Y | Incl. Type | Length 1 | Length 2 |
|---|---|---|---|---|---|---|
| [ 1] | [31] | [ 889] | [967] | 'OA' | '104.7' | '0' |
| [32] | [32] | [1716] | [854] | 'OG' | ' 0.9' | '0' |

Table 4.27: Matrix $R$ shows the results for image 5 (Fig. 4.11).

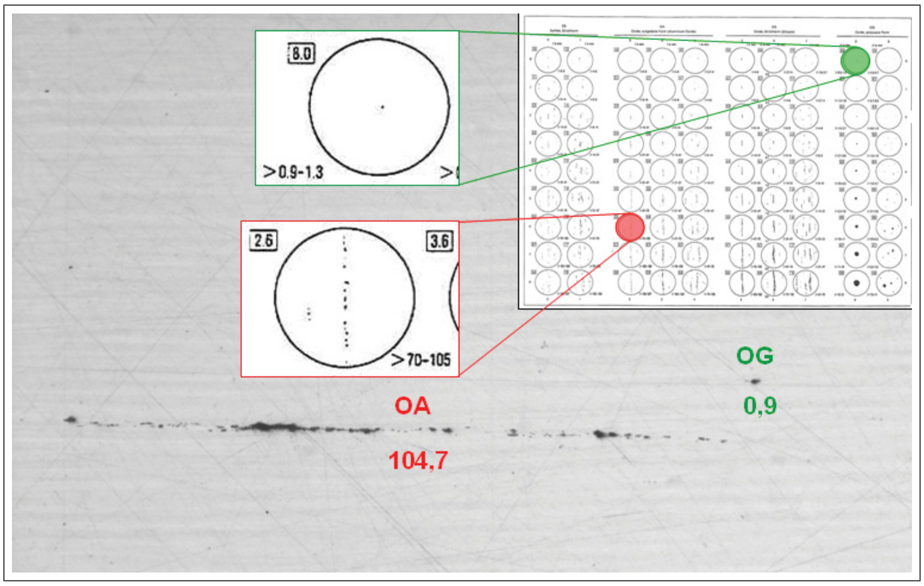
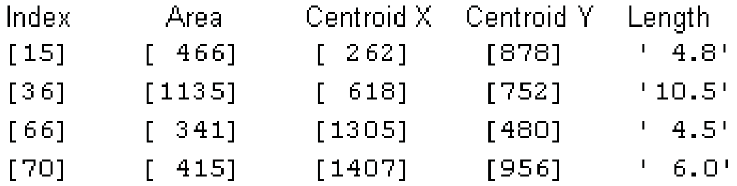


Figure 4.13: Image 5 with type (OA, OG) and length of the inclusions and their position on the DIN 50 602 board. The crucial lengths are: 104.7 (OA), 0.9 (OG).

## 4.2 Classification of Sulfides - SS

The detection of sulfides is much easier than the oxide detection, because there is only one type of sulfides (line-shaped sulfides - SS). To detect sulfides a simple *Thresholding* (see Eq. 3.2) followed by the *regiongrow* function is used. 155 and 185 were used as threshold values for the sample image 6 (Fig. 4.14), which contains only inclusions of the SS type. That means all values between 155 and 185 were set 0, the other values were set 1. After this preprocessing the morphological characters are calculated (like matrix $M$ for oxides) and if the area is smaller than 200 pixels the region is detected as an error. The Matrix $R$ contains the computed results: Column 1 is the index number, column 2 defines the area, column 3 and 4 define the coordinates of the centroid and column 5 defines the length of the inclusion. Table 4.28 and Fig. 4.15 show the results.
To combine the detection of oxides and sulfides just both algorithms are working separately on the image as shown in Fig. 4.16.

Figure 4.14: Image 6: Specimen with inclusions of type SS.



```
Index      Area      Centroid X   Centroid Y   Length
[15]     [ 466]     [ 262]       [878]        ' 4.8'
[36]     [1135]     [ 618]       [752]        '10.5'
[66]     [ 341]     [1305]       [480]        ' 4.5'
[70]     [ 415]     [1407]       [956]        ' 6.0'
```

Table 4.28: Matrix $R$ shows the results for image 6.

Further sample images and the results of the characterization of their inclusions are shown in the Appendix.

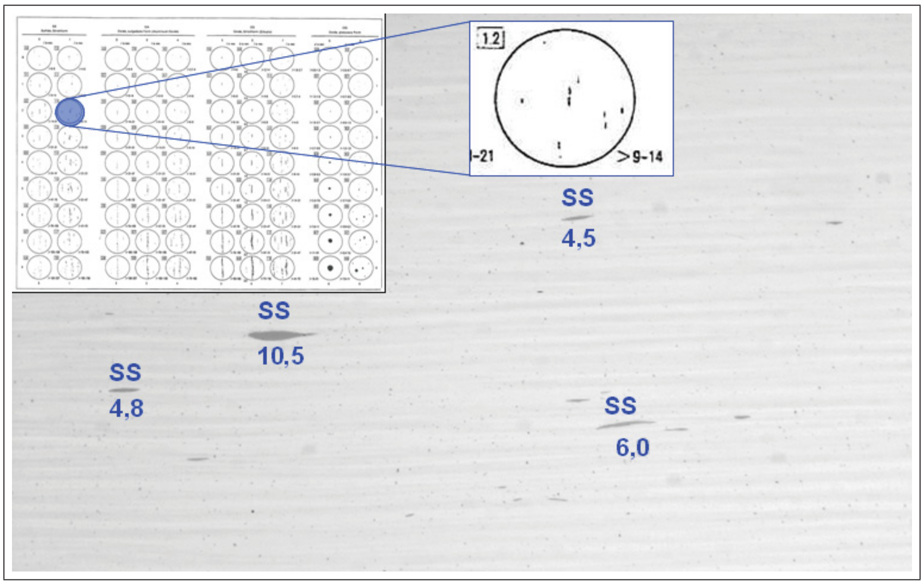


Figure 4.15: Image 6 with type (SS) and length of the inclusions and their position on the DIN 50 602 board. The crucial length is 10.5.

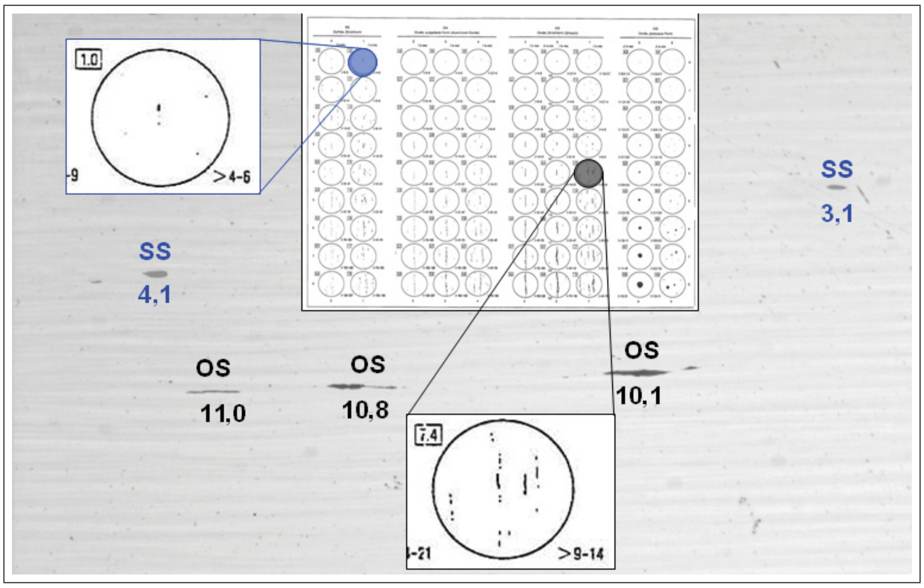


Figure 4.16: Image 7 with type (OS, SS) and length of the inclusions and their position on the DIN 50 602 board. The crucial lengths are: 4.1 (OS), 11.0 (SS), 6.4 (OS).

## 4.3 Statistical Overview

In this work 47 images with a total of 131 inclusions were analyzed and classified with those operations that were presented in the previous sections. Figure 4.17 shows the result of classification. The red bar stands for the overall number of inclusions of the same type. The yellow bar stands for the number of correctly classified inclusions.

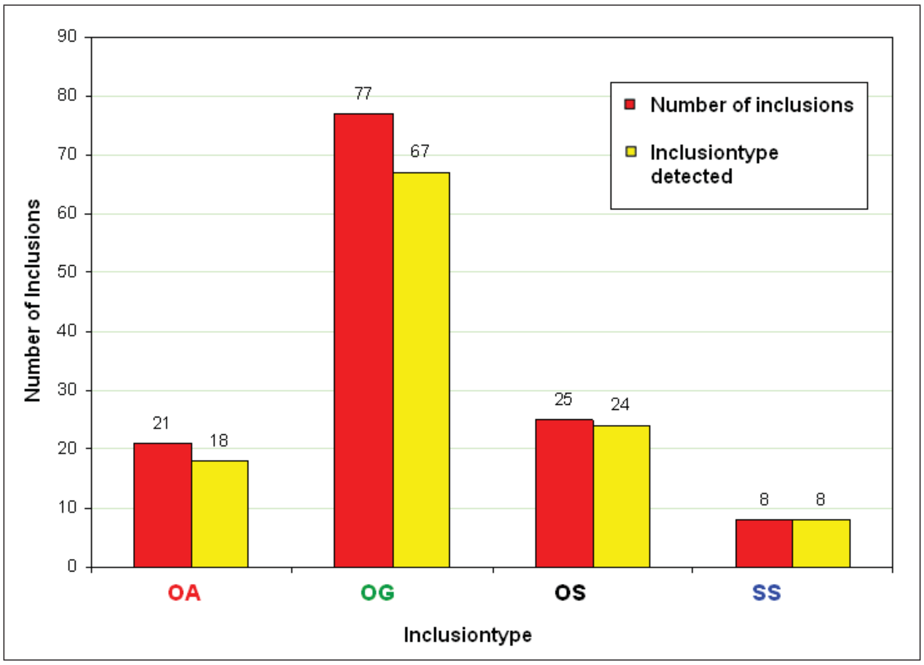


Figure 4.17: Itemizing of the inclusions. The red bar stands for the number of analyzed inclusions. The yellow bar stands for the number of detected inclusions.

The detection of sulfides worked excellent so that 8 out of 8 sulfides were classified correctly. 18 out of 21 dissolved oxides inclusions were classified correctly. With 67 out of 77 globular oxides were classified commonly good. The detection of line-shaped oxides worked well too with 24 out of 25 correctly classified inclusions.

# 5 Conclusion and Outlook

The objective of this work was to study whether or not it is possible to extract and classify given steel inclusions by automated inspection. It was aimed to conclude with the prognosis if a change to automated inspection and classification is useful. For this feasibility study 47 images with a total of 131 inclusions were analyzed and classified using state of the art operations as well as new developed methodologies operations.
The separation of oxides and sulfides worked well, because these inclusions show vastly different gray level values. Problems could appear if errors like scratches or holes have a similar intensity level as sulfides, but to answer this question a further study with more sample images including errors is necessary.
The analysis of different types of oxide was more difficult. A dissolved oxide consists of regions of small globular oxides or oxides which can also be classified as line-shaped oxides. However 18 out of 21 inclusions were classified correctly. A future goal is to improve the connection of the regions. Oxides lose intensity on the end and start sections. That's why a thresholding can produce a small length error. For the improvement of the connective procedure and thresholding more images are necessary.
Globular Oxides were detected commonly good (67 out of 77 ). For oxides with size index 0 the classification is difficult and therefore 200x magnification of images would be better.
The detection of line-shaped Oxides worked well too with 24 out of 25 detected inclusions. Problems can occur only if a line-shaped oxide is a part of a dissolved oxide.

The feasibility study delivered results that show that a change to automated inspection and classification is possible. However it is advisable to further improve processing of the images with more sample images of different inclusion types and errors. To detect errors the textural features for image classification based on *Co-occurrence Matrices* can be an important tool. Improvements on the camera like a higher resolution can also deliver better results.

# 6 Appendix

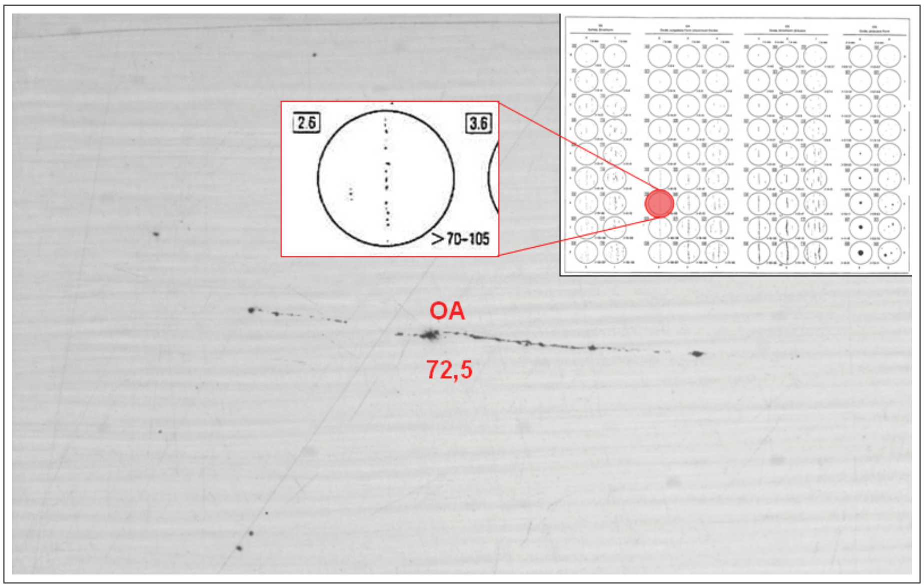


Figure 6.1: Type (OA) and length (72.5) of an inclusion and its position on the DIN 50 602 board.

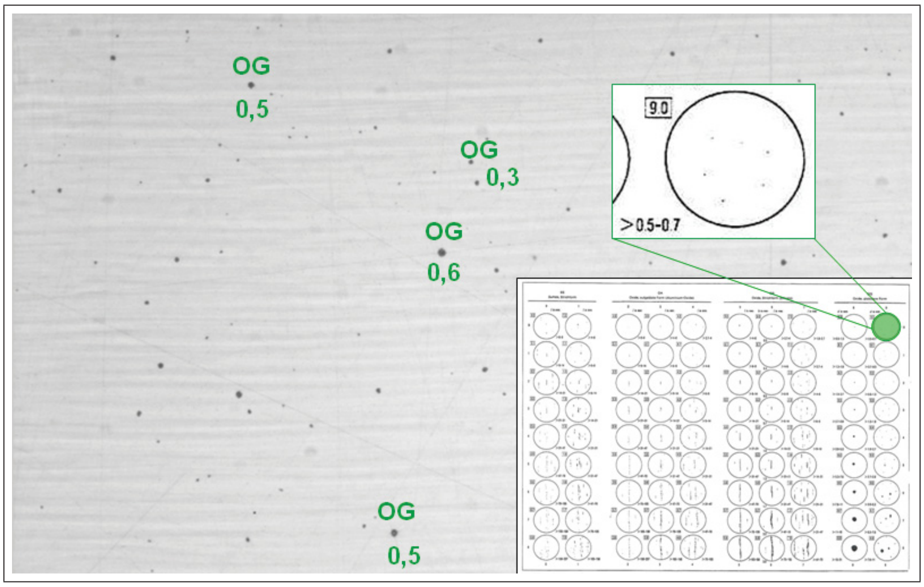


Figure 6.2: Type (OG) and length of inclusions and their position on the DIN 50 602 board. The crucial length is 0.6. The image was taken with a 200x magnitude.

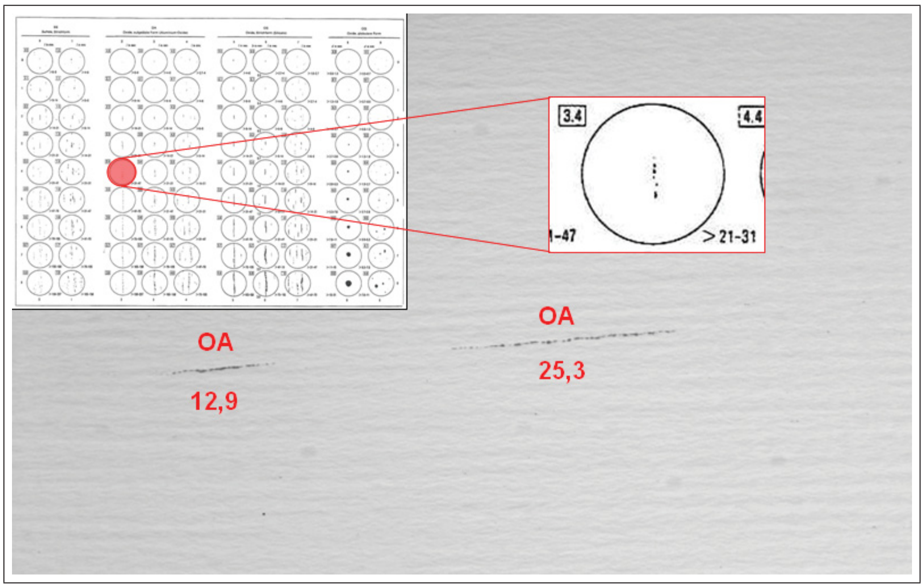


Figure 6.3: Type (OA) and length of inclusions and their position on the DIN 50 602 board. The crucial length is 25.3.

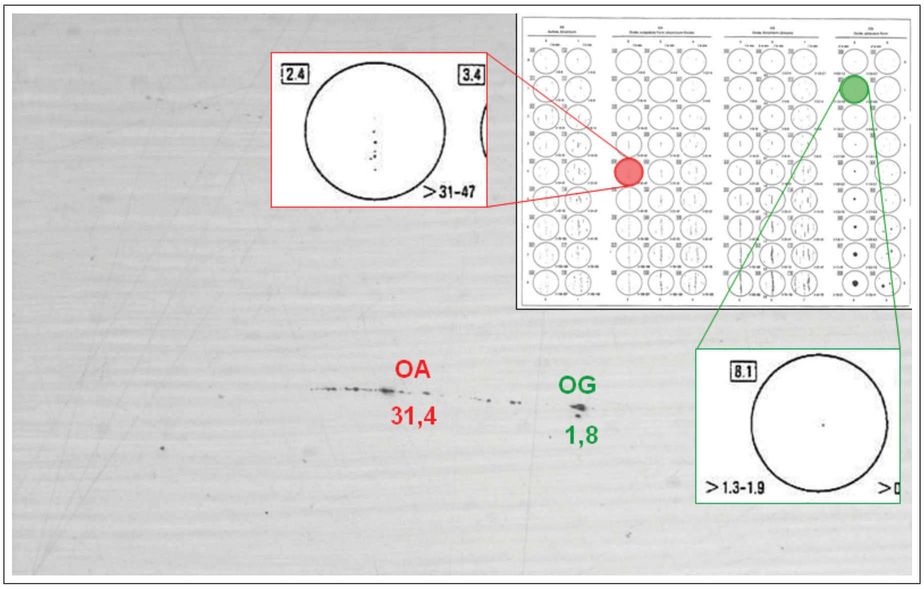


Figure 6.4: Type (OA, OG) and length of inclusions and their position on the DIN 50 602 board. The crucial lengths are 31.4 (OA) and 1.8 (OG).

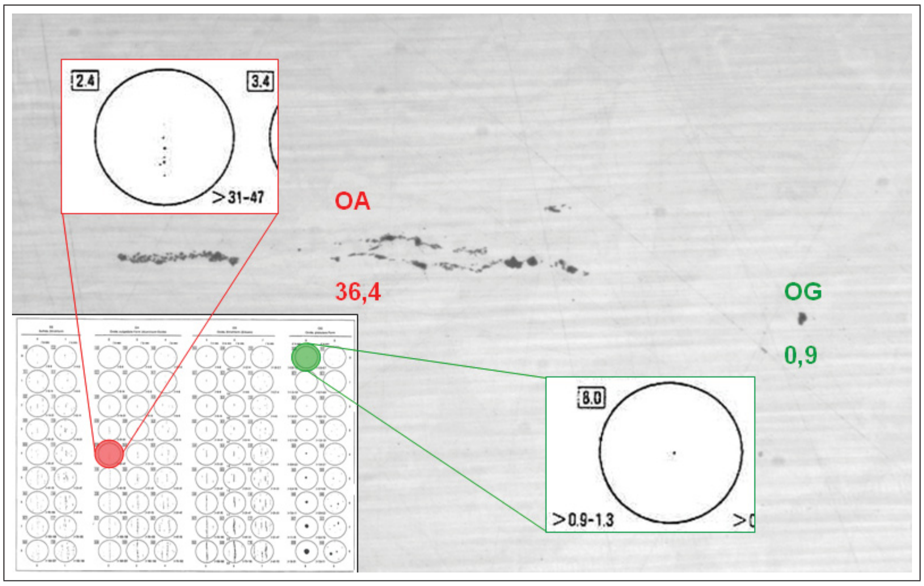


Figure 6.5: Type (OA, OG) and length of inclusions and their position on the DIN 50 602 board. The crucial lengths are 36.4 (OA) and 0.9 (OG). The image was taken with a 200x magnitude.

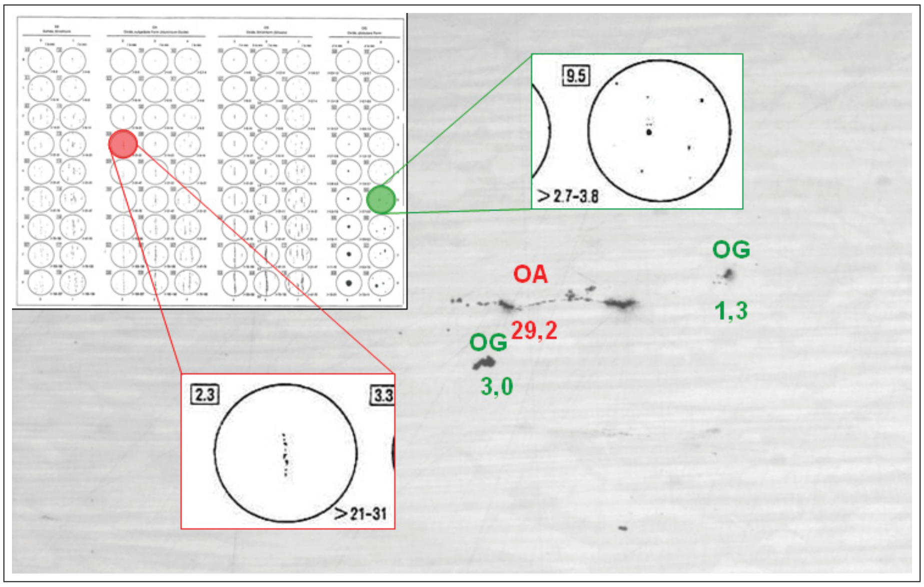


Figure 6.6: Type (OA, OG) and length of inclusions and their position on the DIN 50 602 board. The crucial lengths are 29.2 (OA) and 3.0 (OG).

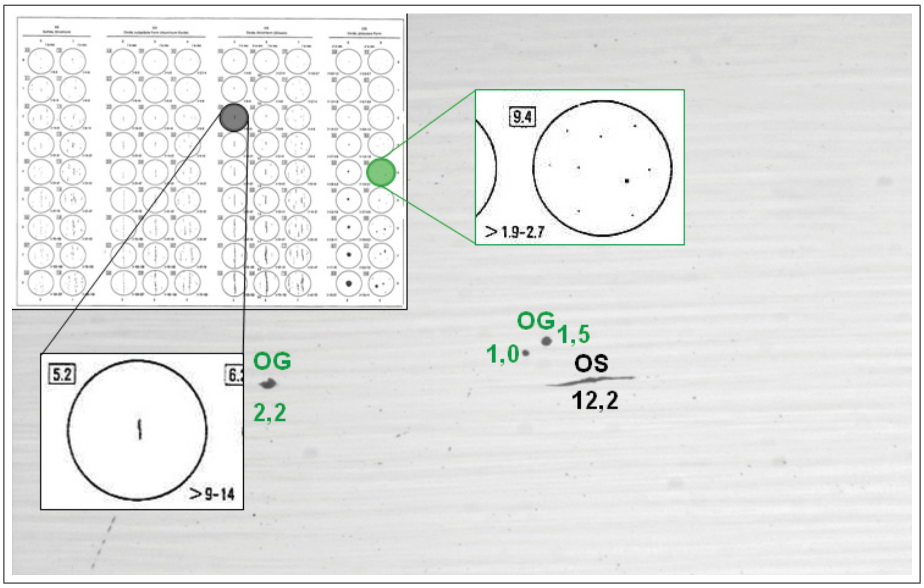


Figure 6.7: Type (OG, OS) and length of inclusions and their position on the DIN 50 602 board. The crucial lengths are 2.2 (OG) and 12.2 (OS).

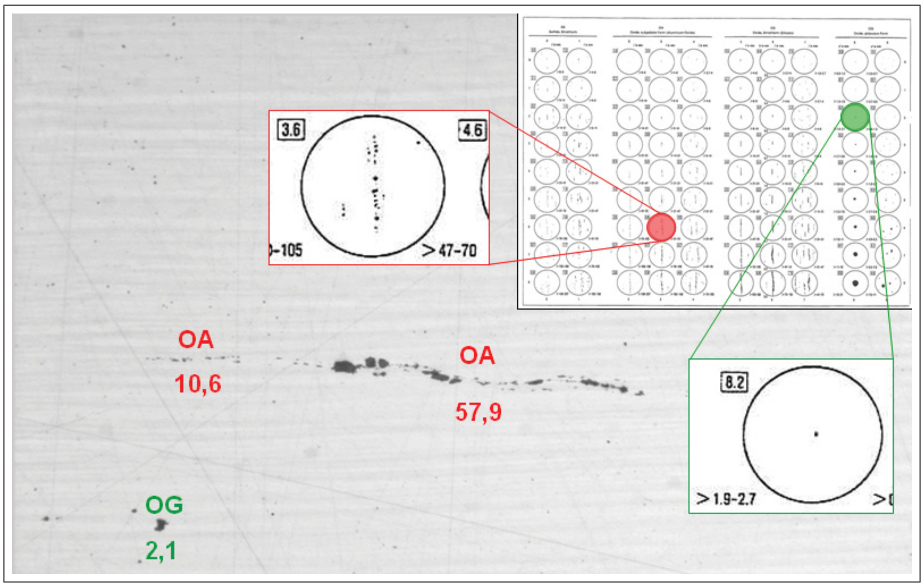


Figure 6.8: Type (OA, OG) and length of inclusions and their position on the DIN 50 602 board. The crucial lengths are 57.9 (OA) and 2.1 (OG).

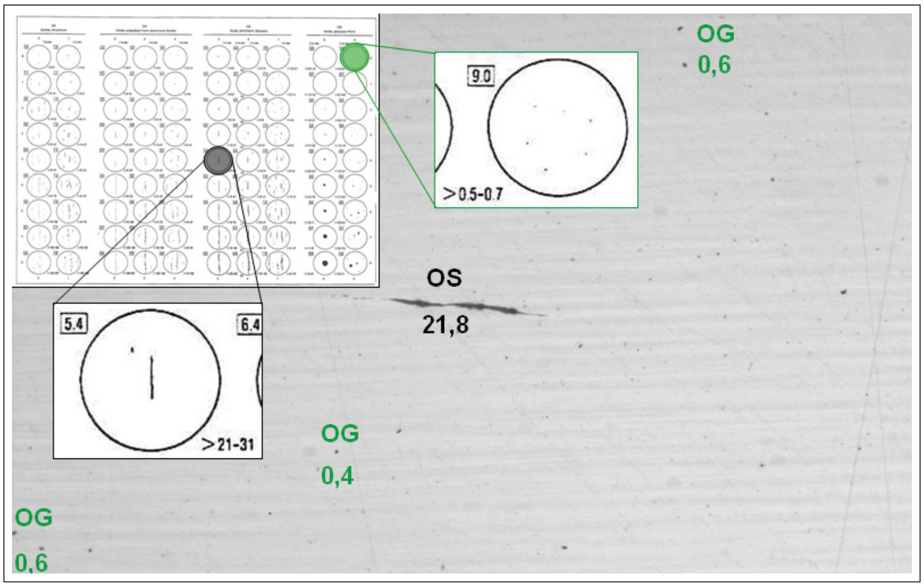


Figure 6.9: Type (OG, OS) and length of inclusions and their position on the DIN 50 602 board. The crucial lengths are 0.6 (OG) and 21.8 (OS).

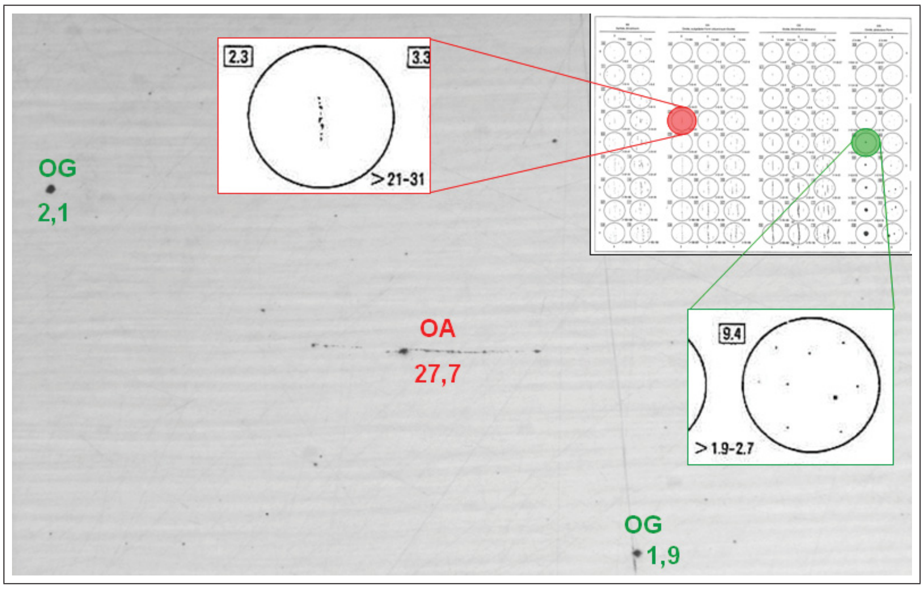


Figure 6.10: Type (OA, OG) and length of inclusions and their position on the DIN 50 602 board. The crucial lengths are 27.7 (OA) and 2.1 (OG).

# Bibliography

[1] B. Jähne. *Digital Image Processing*. Springer Verlag, 2002.

[2] S.L. Eddins R.C. Gonzalez, R.E. Woods. *Digital Image Processing Using MATLAB*. Prentice Hall, 2003.

[3] K.I. Laws. *Textured Image Segmentation*. PhD thesis, University of Southern California, 1980.

[4] E.L. Hall R. Sutton. Texture Measures for Automatic Classification of Pulmonary Disease. *IEEE Transactions on Computers*, pages 667–676, 1972.

[5] H.M. Aus H. Harms, U. Gunzer. Combined Local Color and Texture Analysis of Stained Cells. *Computer Vision, Graphics, and Image Processing*, (33):364–376, 1986.

[6] S.N. Srihari D. Wang. Classification of Newspaper Blocks Using Texture Analysis. *Computer Vision, Graphics, and Image Processing*, (47):327–352, 1989.

[7] J.S. Payne, T.J. Stonham, D.Patel. Document segmentation using texture analysis. In *Pattern Recognition, 1994. Vol. 2 - Conference B: Computer Vision  Image Processing., Proceedings of the 12th IAPR International. Conference*, volume 2, pages 380–382. International Association for Pattern Recognition, 1994.

[8] G. Marcone, G. Martinelli, P. D'Angelo. Automatic face image recognition: an accurate eye locationprocedure. In *Digital Signal Processing Proceedings, DSP 97.*, pages 515 – 519. 13th International Conference on Volume 2, 1997.

[9] M. Storer, M. Urschler, H. Bischof, J.A. Birchbauer. Face Image Normalization and Expression/Pose Validation for the Analysis of Machine Readable Travel Documents. In *32nd Workshop of the Austrian Association for Pattern Recgnition, Linz, 2008*, pages 29–40. AAPR/OAGM, 2008.

[10] F. Alonso-Fernandez, J. Fierrez-Aguilar, J.Ortega-Garcia. An enhanced Gabor filter-based segmentation algorithm for fingerprint recognition systems. In *Proceedings of the 4th International Symposium*. Image and Signal Processing and Analysis, 2005.

[11] L. Wang, M. Dai, G. Geng. Fingerprint image segmentation by energy of Gaussian-Hermite moments. In *Advances in biometric person authentication*, pages 414–423. SINOBIOMETRICS 2004 : Chinese conference on biometric recognition Nr. 5, 2004.

[12] R.D. Joseph E.M. Darling. Pattern recognition from satellite altitudes. *IEEE Trans. Syst. Sci. Cybern.*, SSC-4:38–47, 1968.

[13] E. Lopez-Ornelas, F. Laporterie-Dejean, G. Flouzat. Satellite image segmentation using graph representation and morphological processing. In *Image and signal processing for remote sensing IX* , pages 104–113. Image and signal processing for remote sensing. Conference Nr.9, Barcelona, 2004.

[14] W. Niehsen. Open Problems in Computer Vision. In *36. Heidelberger Bildverarbeitungsforum*, 2008.

[15] A. Davari J. Chittooru, R. Munasinghe. Edge detection and segmentation for machine vision. *Proceedings of the Thirty-Seventh Southeastern Symposium on System Theory, 2005. SSST '05*, pages 457–461, 2005.

[16] M.M. Trivedi T. Gandhi. Pedestrian protection systems: Issues, survey, and challenges. *IEEE Trans. Intell. Transportation Systems*, 8:413–430, 2007.

[17] C. Y. Suen M. M. Zadeh, T. Kasv. Submitted to spie's intelligent system and automated manufacturing 1 localization and recognition of traffic signs for automated vehicle control systems.

[18] J. Canny. A Computational Approach To Edge Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:679–714, 1986.

[19] H. Kaizer. A Quantification of textures on aerial photographs. *Boston Univ. Res. Lab., Boston, Mass.*, 1955.

[20] R. Chevallier, A. Fontanel, G. Grau, M. Guy. 11th Congr. Int. Photogrammetric. In *Application du filtrage optique a l'etude des photographies aeriennes*, 1968.

[21] V. Fish J. Hawkins R. Loewe R. Bixby, G. Elerding. Natural image computer. *Aeronutronic Division, Philco-Ford corp., Newport Beach, Calif., Final Tech. Rep.*, 1, 1967.

[22] R.M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, pages 610–621, 1973.

[23] C. Schmid K. Mikolajczik. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615 – 1630, 2005.

[24] K. Amlacher, L. Paletta, P. Luley, G. Fritz, A. Almer. Geo-Indexed Object Recognition in Urban Scenarios. In *32nd Workshop of the Austrian Association for Pattern Recgnition, Linz, 2008*, pages 9–18. AAPR/OAGM, 2008.

[25] L. Tesar, D. Smutek, A. Shimizu, H. Kobatake. Medical image segmentation using cooccurrence matrix based texture features calculated on weighted region. In *Proceedings of the third conference on IASTED International Conference: Advances in Computer Science and Technology*, pages 243–248. International Association for Science and Technology for Development, 2007.

[26] J.W. Sayre J.G. Goldin D.R. Aberle M.F. McNitt-Gray, N. Wyckoff. The effects of co-occurrence matrix based texture parameters on the classification of solitary pulmonary nodules imaged on computed tomography. *Comput. Med. Imaging Graph.*, pages 339–348, 1999.

[27] A.F.C. Infantosi C.M. Azevedo A.V. Alvarenga, W.C.A. Pereira. Complexity curve and grey level co-occurrence matrix in the texture evaluation of breast tumor on ultrasound images. *Medical Physics*, 34(2):379–387, 2007.

[28] R. Kwitt, A. Uhl. Color wavelet cross co-occurrence matrices for endoscopy image classification. In *Communications, Control and Signal Processing, 2008. ISCCSP 2008. 3rd International Symposium*, pages 715 – 718, 2008.

[29] K. Lee, S.H. Jeon, B. Kwon. Map Asia Conference. In *Communications, Control and Signal Processing, 2008. ISCCSP 2008. 3rd International Symposium*, 2004.

[30] G. Lohmann. Co-occurrence-based analysis and synthesis of textures. In *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision Image Processing., Proceedings of the 12th IAPR International Conference*, pages 449 – 453, 1994.

[31] J. Furst D.S. Raicu A.S. Kurani, D. Xu. Co-occurrence matrices for volumetric data.

[32] C.J. Taylor J. Haslam T.F. Cootes, A. Hill. The use of active shape models for locating structures in medical image. *Image Vision Comput.*, (12), 1994.

[33] M. Petrou I.Ng K.Y. Song, J. Kittler. Chromato-structural approach towards surface defect detection in random textured images. *Proceedings, Machine Vision Applications in Industrial Inspection II*, 1994.

[34] H. Nieman. *Pattern Analysis and Understanding.* Springer Verlag, 1990.

[35] H. Freeman. On the encoding arbitrary geometric configuration. *IRE Trans. Electronic Comput.*, EC-10(2):260–268, 1961.

[36] M.M. Galloway. Texture analysis using gray level run lengths. *Comput. Graphics Image Process. 4*, pages 172–179, 1975.

[37] S. Rüger P. Horwath. Evaluation of Texture Features for Content-based Image Retrieval. *Department of Computing, Imperial College London*, 2003.

[38] P.G. Sevilla M. Petrou. *Image Processing: Dealing with texture.* Wiley, 2006.

[39] M. Petrou V. Kovalev. Multidimensional Co-occurrence Matrices for Object Recognition and Matching. *Graphical Models and Image Processing*, 58(3):187–197, 1996.

[40] Y.S. Boundar V. Kovalev, M. Petrou. Using orientation tokens for object recognition. *Pattern Recognition Letters 19*, pages 1125–1132, 1998.

[41] M.K. Hu. Visual Pattern Recognition by Moment Invariants. *IRE Trans. Info. Theory*, IT-8:179–187, 1962.

# List of Tables

# List of Figures