



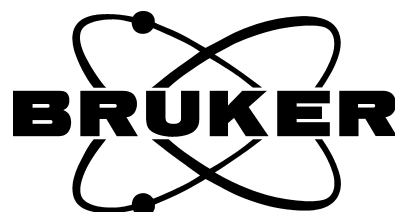
OPUS

Spectroscopy Software

Version 6

User Manual

OBJECT



© 2006 BRUKER OPTIK GmbH, Rudolf-Plank-Str. 27, D-76275 Ettlingen, www.brukeroptics.com

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means including printing, photocopying, microfilm, electronic systems etc. without our prior written permission. Brand names, registered trade marks etc. used in this manual, even if not explicitly marked as such, are not to be considered unprotected by trademarks law. They are the property of their respective owner.

The following publication has been worked out with utmost care. However, Bruker Optik GmbH does not accept any liability for the correctness of the information. Bruker Optik GmbH reserves the right to make changes to the products described in this manual without notice.

This manual is the original documentation for the OPUS spectroscopic software.

Table of Contents

- 1 Introduction.....1**

- 2 Technical Requirements.....3**
 - 2.1 Software Requirements 3
 - 2.2 Hardware Requirements 3

- 3 Image Processing Functions.....5**
 - 3.1 Compute Grayscale Image..... 5
 - 3.2 Median 6
 - 3.3 Negative..... 6
 - 3.4 Stretch Histogram 7
 - 3.5 Binarization 7
 - 3.6 Erode 9
 - 3.7 Dilate 10
 - 3.8 Opening 10
 - 3.9 Find Regions 11
 - 3.10 Measuring Spectra of the Found Regions 12

1 Introduction

The OPUS/OBJECT package contains a set of image processing functions for detecting objects within digital images. This software can be best used in connection with the OPUS/VIDEO software and an x/y-stage controlled by the OPUS/MAP software. Using the OPUS/MAP software you can interactively define an interesting sample area. Then the OPUS/VIDEO software automatically generates an overview image of that sample area which is assembled from various single image frames at different stage positions. The OPUS/OBJECT package can then be used to find objects of interest in the overview image, to extract their size and position and to measure IR-spectra of the found objects by moving the x/y-stage to corresponding position.

2 Technical Requirements

2.1 Software Requirements

The OPUS/OBJECT software requires the OPUS/VIDEO software (to control the frame grabber board and to produce the VIS-images) and the OPUS/MAP software (to control the computer-controlled x/y-stage).

2.2 Hardware Requirements

The analysis of large images requires a high RAM and hard-disk storage capacity depending on the size of the overview image area. The following example illustrates the storage capacity requirements.

If you scan a sample area of 1cm^2 using the IRScopeII or Hyperion microscope (with 4x VIS-objective but without an optical adapter), one image frame covers an area of 0.848mm in x and 0.636mm in y. The generated overview image is composed of 192 individual image frames (12 frames in x-direction and 16 frames in y-direction) with each frame having a size of 1.3 Megabyte. That means that the resulting RGB-overview image requires a storage capacity of 250 Megabyte. If you use the 15x IR objective instead of the 4x objective, one image frame covers only an area of 0.29mm in x-direction and 0.217mm in y-direction. The resulting overview image is then composed of 1,610 image frames (35 frames in x-direction, 46 frames in y-direction) requiring a storage capacity of 2 Gigabyte. During the image processing two grayscale images are computed and stored in addition to the original RGB-image. They both increase the amount of image information by about 70% of the original RGB-image.

As images of these sizes cannot be stored in RAM, the software works disk-interactively: it is assumed that an image having the same size in x-direction as the overview image and consisting of one frame height in y-direction can be stored in RAM. In case of an image generated by the 4x VIS-objective this means that 12 image frames ($12 \times 1.3 = 15.6$ Megabyte) must be stored in RAM, while in case of the 15x IR-objective this would be 35 image frames requiring a storage capacity of 46 Megabyte. These image sizes can still be handled if RAM has the recommended disk space of 256 Megabyte. To avoid swapping do not use a RAM with a lower storage capacity.

The software enables you to store the images in JPEG-format, which compresses the data significantly and thus reduces the necessary disk space. However, this compression is not loss-free, i.e. it slightly modifies the data and thus cannot be used as input images for operations like *Erode*, *Dilate* or *Find Regions* which require an 'exact' binarized image. Therefore, even if you select

JPEG-format, the software automatically switches to BMP-format, if required. Note that the compression does not reduce the disk space needed in RAM, because JPEG-images are decompressed to BMP-format and require the same disk space in RAM as BMP-images.

As the images require a high storage capacity the available free disk space should be as large as possible. In addition, the possibility of saving data on a server or a CD-writer should be taken into consideration right from the beginning.

Finally, storing, reading and computing such large amounts of data requires a fast hard-disk and powerful CPU.

3 Image Processing Functions

The *Image processing* functions can be accessed via the pop-up menu of the overview window, provided you have registered your OPUS/OBJECT package. (See OPUS Reference Manual for details.)

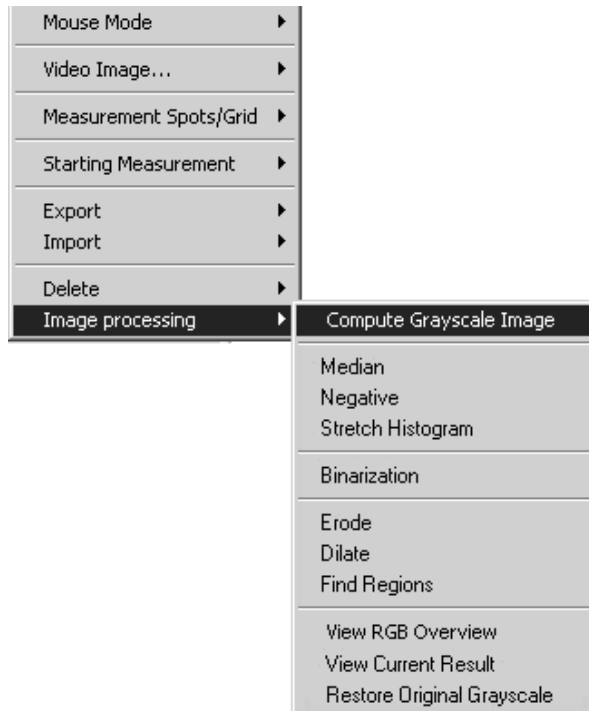


Figure 1: Image Processing - Pop-up Menu

3.1 Compute Grayscale Image

First, a grayscale image is created using the *Compute Grayscale Image* function. After the computation is completed, the grayscale image instead of the original RGB-image is displayed. You can switch between the RGB-Overview image and the result of the last image processing operation using the *View RGB Overview* and *View Current Result* functions.

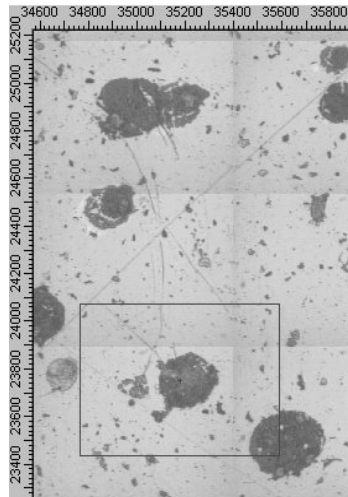


Figure 2: Grayscale Image of the Sample

3.2 Median

This function eliminates extreme intensity values at certain pixels by moving a 5x5 mask over the image, sorting the corresponding 25 intensity values in a linear array, selecting the 13th intensity value and substituting the original pixel at the center of the 5x5 mask by this value. Use this function if a few pixels, being surrounded by normal pixels, are completely wrong and have to be eliminated.

3.3 Negative

This function inverts the intensities of an image and thus produces the negative of the original grayscale image. Use this function if the objects of interest are darker than the background, because the algorithm for detecting objects assumes that the objects are lighter than the background.

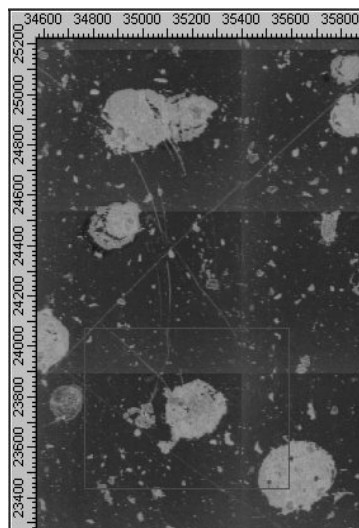


Figure 3: Negative Image of the Sample

3.4 Stretch Histogram

If the pixel intensities in an image do not cover the full dynamic range from 0 to 255 (e.g. from 80 to 255), the negative image will be too dark to see details. So, if the total contrast of the image is too low, you can use the *Stretch Histogram* function to change the intensity distribution in such a way that the whole dynamic range is covered. This operation is only necessary, if you want to see details in the image by visual inspection. If you work with automatic object detection, this step can be omitted.

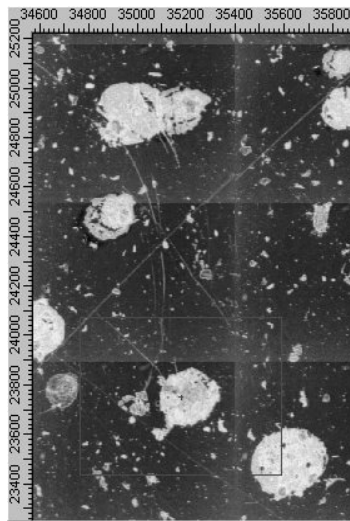


Figure 4: Result after using *Stretch Histogram*

3.5 Binarization

This function assigns to all image pixels, of which the intensity is above a certain threshold, the maximum intensity (255, white), while those pixels with an intensity below a certain threshold are assigned to the minimum intensity (0, black). The threshold is set interactively in the following window which opens after *Binarization* is started:

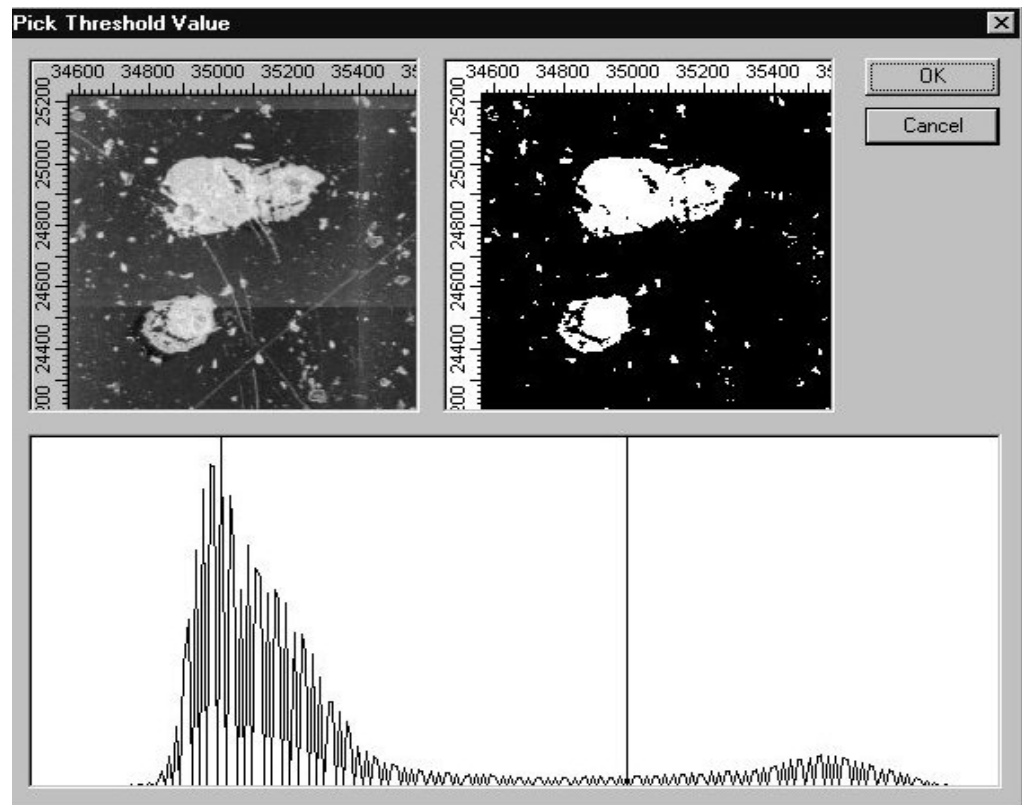


Figure 5: Binarization Window - Interactive Threshold Setting

At the bottom of the window the image histogram (i.e. the intensity distribution of the image) is shown. The x-axis lists all possible intensity values of the gray-scale image from 0 to 255, the y-axis displays the number of pixels having a certain intensity value x . The aim of the binarization is to show all interesting objects in white on a black background. This is only possible, if the objects are lighter than all other areas of the image. Ideally, the histogram has two peaks: the peak at lower intensities represents the dark background while the peak at higher intensities arises from the lighter objects. To separate the objects from the background you have to define an intensity value between both peaks. This can be done by moving the vertical line with the mouse to the right or to the left and watching the effect on the right-hand image. The binarized image can be further processed using the *Erode* and *Dilate* functions to remove areas that are too small.

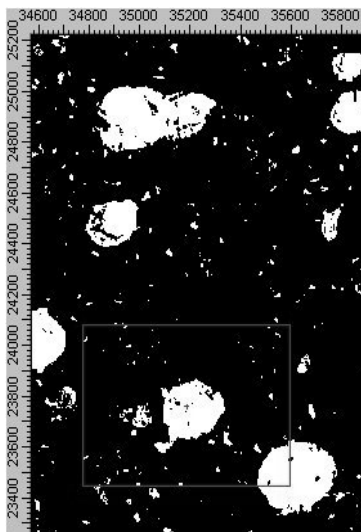


Figure 6: Image after Binarization

3.6 Erode

This function can only be applied to a binarized image. It eliminates small objects by moving a 5x5 rectangle over the image and deleting the pixel at the central point of the rectangle, if there is at least one black pixel inside the 5x5 rectangle. All regions consisting of less than 25 pixels are deleted in this way. Thin connections between larger objects are also erased and the size of all objects is reduced by at least 2 pixels on each boundary.

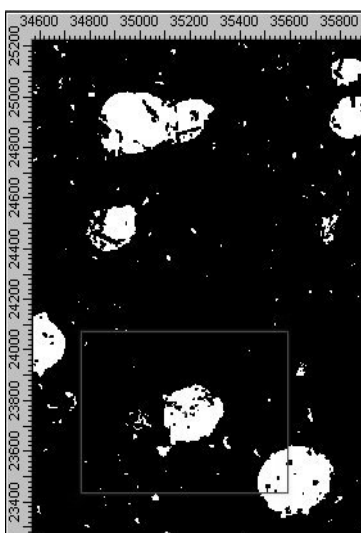


Figure 7: Image after Erosion

3.7 Dilate

This function is the counterpart to the *Erode* function. It can also be applied to binarized images only. The function produces just the opposite effect by moving a 5x5 rectangle over the image. Wherever the central pixel of the rectangle is placed over a white 'object'-pixel, all 5x5 surrounding pixels are converted into white pixels. By doing so, small holes are filled and the white regions are re-enlarged to their original size.

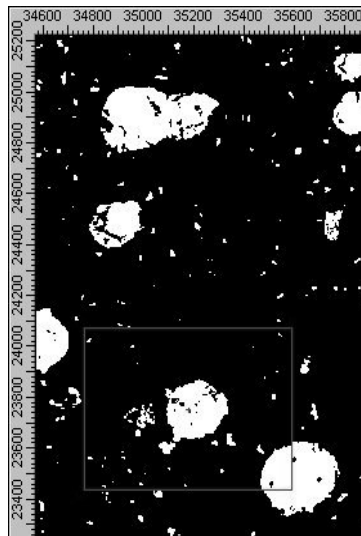


Figure 8: Image after Dilation

3.8 Opening

Normally, *Erode* is followed by *Dilate* in order to retain the size of the remaining objects. The combination of these two operations is called *Opening*: first it completely eliminates all small objects by the erosion and then it fills small gaps and re-enlarges the remaining objects to their original size. The effects of an *Erode* and *Dilate* operation become clearer if the image is zoomed until individual pixels become visible. The following figure illustrates the effects of an *Opening* operation.

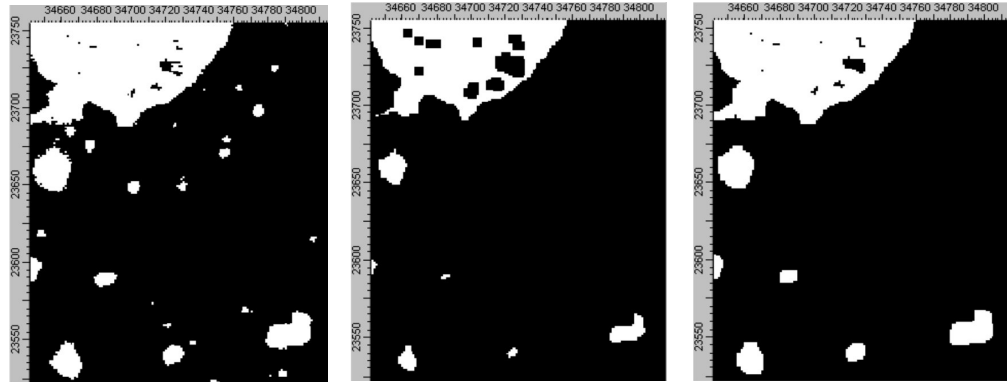


Figure 9: Effect of an Opening operation

3.9 Find Regions

This function searches the binarized image for white areas and determines their position (=center of gravity) and size. Typically, most samples contain a large number of tiny objects with the size of a few pixels. In most cases these tiny objects are not of interest as they may represent undesired information like noise or dust particles. After having found all regions, a dialog box pops up which allows the user to discard all objects smaller than a user-defined minimum size.

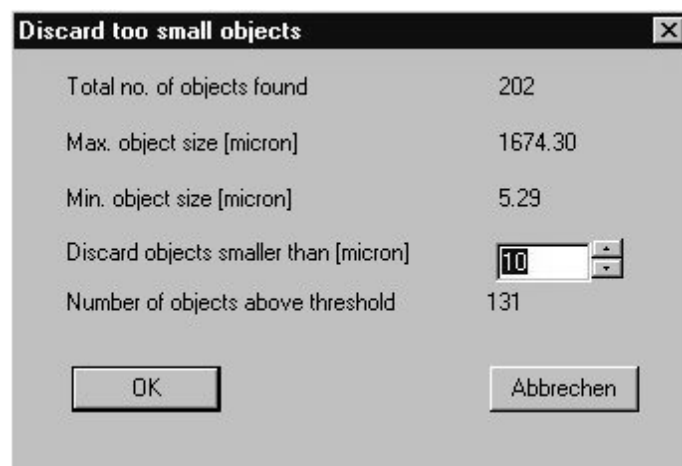


Figure 10: Image Statistics and Defining a Threshold

If you click on the *OK* button, the found regions are marked by circles. In addition, the regions representing the objects (formerly white regions) are combined and represented in false color with the underlying image:

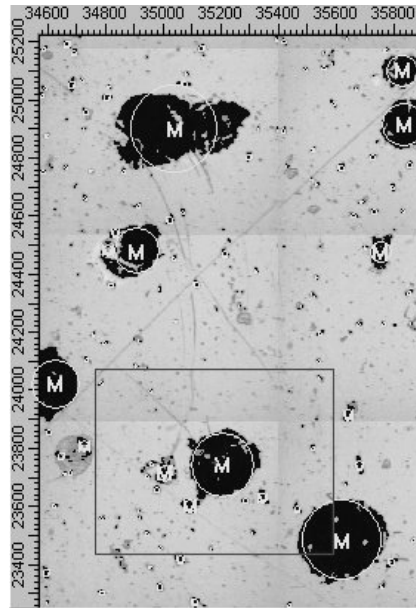


Figure 11: Found Objects for Measurement

3.10 Measuring Spectra of the Found Regions

As long as the video image window is open, all found regions can be measured automatically using the *Measure Marked Positions* function of the *Video* menu. This measurement function works in the same way as if the regions were defined interactively. Before starting the sample measurement, a background spectrum must be acquired. Especially, if a larger number of spots has to be measured, it is recommended to repeat the background measurement after every n^{th} sample measurement. For this purpose, you have to define a x/y-stage position for the background spectrum. The measurement can be stopped by clicking on the *Stop* button. The circles around those spots which have been measured adopt the color of the corresponding spectrum and the letter 'M' changes to the running number of the spectrum in the 3D file. A tick mark indicates that the spot has been measured.

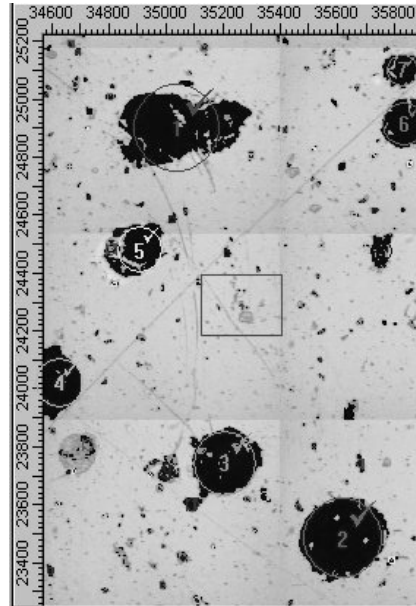


Figure 12: After the Measurement

Index

Numerics

3D file 12

B

Background measurement 12

Binarization 7, 8

Binarized image 3, 8, 10, 11

BMP-format 4

D

Dilate 3, 8, 10

E

Erode 3, 8, 9, 10

Erosion 9, 10

F

False color 11

Find regions 3, 11

Frame 3

Frame grabber board 3

G

Grayscale image 5, 6, 8

H

Histogram 7, 8

I

Intensity 6, 7, 8

Intensity distribution 7, 8

J

JPEG-format 3, 4

M

Measure marked positions 12

Median 6

N

Negative 6

Negative image 7

O

Opening 10

Overview image 1, 3

Overview window 5

R

RAM 3

RGB-image 3, 5

RGB-overview image 3, 5

S

Sample measurement 12