TU Delft
Delft University of Technology

Department of Civil Engineering
Hydraulic and Geotechnical Engineering Division
Hydromechanics Section

Segmentation of silt particles from exposures with background by use of second derivative

J. de Graaff and R.E. Slot

report no. 5-95, November 1993

CONTENTS

APPENDICES:

1. INTRODUCTION

Recently a lot of research is being done on cohesive sediment. It plays a major role in the shoaling of harbours and waterways, and in some serious environmental problems. To predict cohesive sediment transport, information is needed about the distributions of size and settling velocities. Many methods exist to determine sizes of suspended particles, but most are not applicable to cohesive sediment flocs, because of their fragility. If not at sampling, the flocs break at the subsequent analysis by for example the Coulter Counter or the pipet method. In case of analysis by the Owen tube another problem occurs next to the floc break up at sampling: the long duration of the analysis leads to additional flocculation and causes the measured distribution to be even more unrealistic.

To solve these problems, exposures are made by underwater cameras, which give instantaneous information about the undisturbed samples. From one exposure the floc sizes can be determined, and from two successive exposure with known time between them, the settling velocities can be determined.

So far, the analysis of exposures of flocs was mainly done by hand. Image processing by computer provides a way to do this automatically. It saves time, and consequently more flocs can be analyzed, leading to more representative distributions.

The subject of this report is the development and testing of an image processing program to distinguish the objects with use of second derivative method. This method is developed and is compared with other methods. The program is applied to digitized exposures, as can be made by a framegrabber. The framegrabber converts a recording on tape or from a ccd camera into a matrix of digits, the value of each digit representing the brightness of the corresponding pixel. From this grey value image, the image processing program has to distinguish the relevant objects, in other words, make a binary image, consisting of object pixels and non object pixels. This is quite complicated, due to inevitable interferences on the exposures like background features and shadow effects. After producing the binary image, the next program determines particle sizes and calculates the mean size and spread, the measure of the broadness of the distribution.

This report describes the problems that are met when segmenting objects from a background (chapter 2), the mathematical methods to overcome them (chapter 3), some tests (chapter 4), the results of these tests (chapter 5) and some conclusions (chapter 6). The tests have been done on exposures with reference objects (ideal objects and background). The results are also visualized in the appendices.

2. PROBLEMS

Several features of the exposures of cohesive sediment flocs disturb the images of the flocs and cause difficulties when determining the exact shapes of the relevant flocs:

(1) - the exposures do not only consist of objects but also of varying backgrounds;
(2) - the edges of the objects are not equally sharp at every place;
(3) - the brightness of the objects varies;
(4) - shadow effects occur, because of illumination from aside;
(5) - some objects have holes;
(6) - objects at the edge of the image are not completely visible;
(7) - the objects overlap each other.

Problems (1) to (4) concern segmentation of the objects from the background, and are treated in chapter 3.1. Problems (5) to (7) are successively treated in chapter 3.2 to 3.4.

3. METHODS

3.1 Segmentation

The problems (1) up to and including (4) are all about distinguishing the objects from the background. These problems can be solved by using the TCLi-package as follows: first uniform or minmax filtering for correction on shading and second isodata threshold for segmentation. However, these methods fail in some cases. Therefore a program using the second derivative of the brightness has been developed.
Starting with the least complicated method of segmentation direct from the image, and moving to more complicated ones the use of the second derivative is supported, in comparision with the use of uniform filter + isodata threshold and minmax filter + isodata threshold.

Segmentation direct from the image.
Because objects have different brightness and because of the presence of varying background, accurate segmentation direct from the image is difficult:

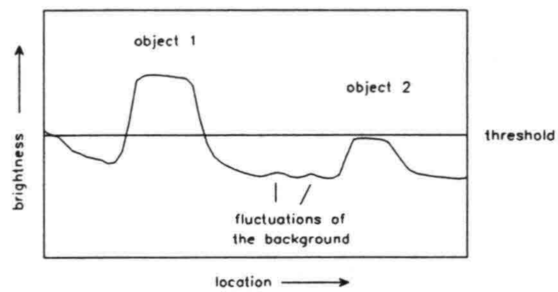


fig. 3.1. A plot from the image consisting a bright and a weak object and background.

By lowering a detection threshold in cases such as shown in fig. 3.1 object number 1 is first detected, second some background is detected on the left; only by further lowering the threshold object number 2 is detected.

Segmentation from the first derivative of the brightness.
Boundaries between objects and background give extreme values in the first derivative.

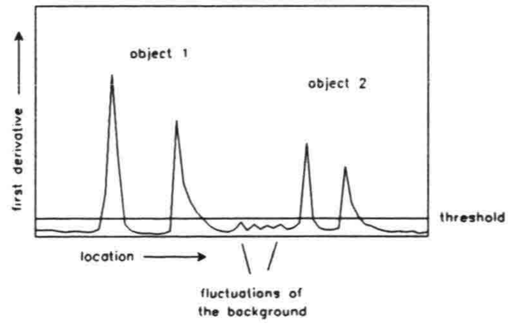


figure 3.2. A plot of the first derivative of brightness of the image in fig. 3.1 (absolute values).

Only peaks of the two objects are detected. There is no problem with different brightness and with the background. After segmentation, the binary image shows boundaries between objects and background as thick lines, which are rings in case the objects are dots. The boundary of the object is defined on the place where there is a maximum gradient of the brightness (first derivative has a peak or second order derivative is equal to zero). In case of a symmetric peak, the real boundaries are found in the middle of the thick lines, the skeletons (fig. 3.3).

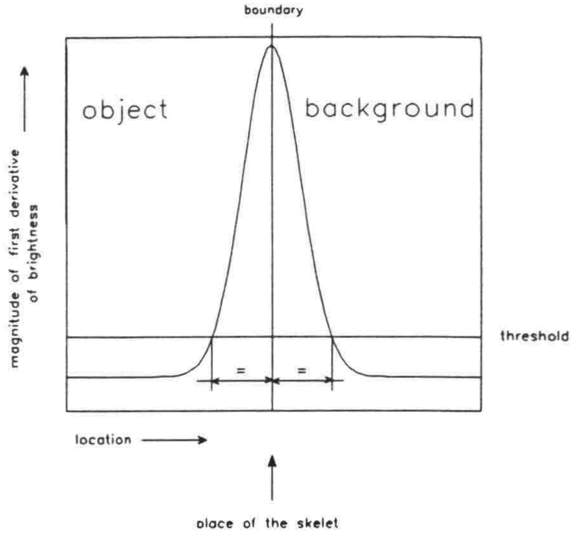


figure 3.3. In case of a symmetric peak the top, which indicates the boundary between object and background, coincides with the skeleton.

Hence the boundaries between objects and background are rather easily determined by the proces of skeletonisation. After filling up the closed spaces formed by the boundary lines, the bitplane shows objects and background. However, this method can not be used if the peaks are not symmetric.

If the first derivative is asymmetric, for example caused by a shadow effect, the next problem occurs, fig. 3.4:

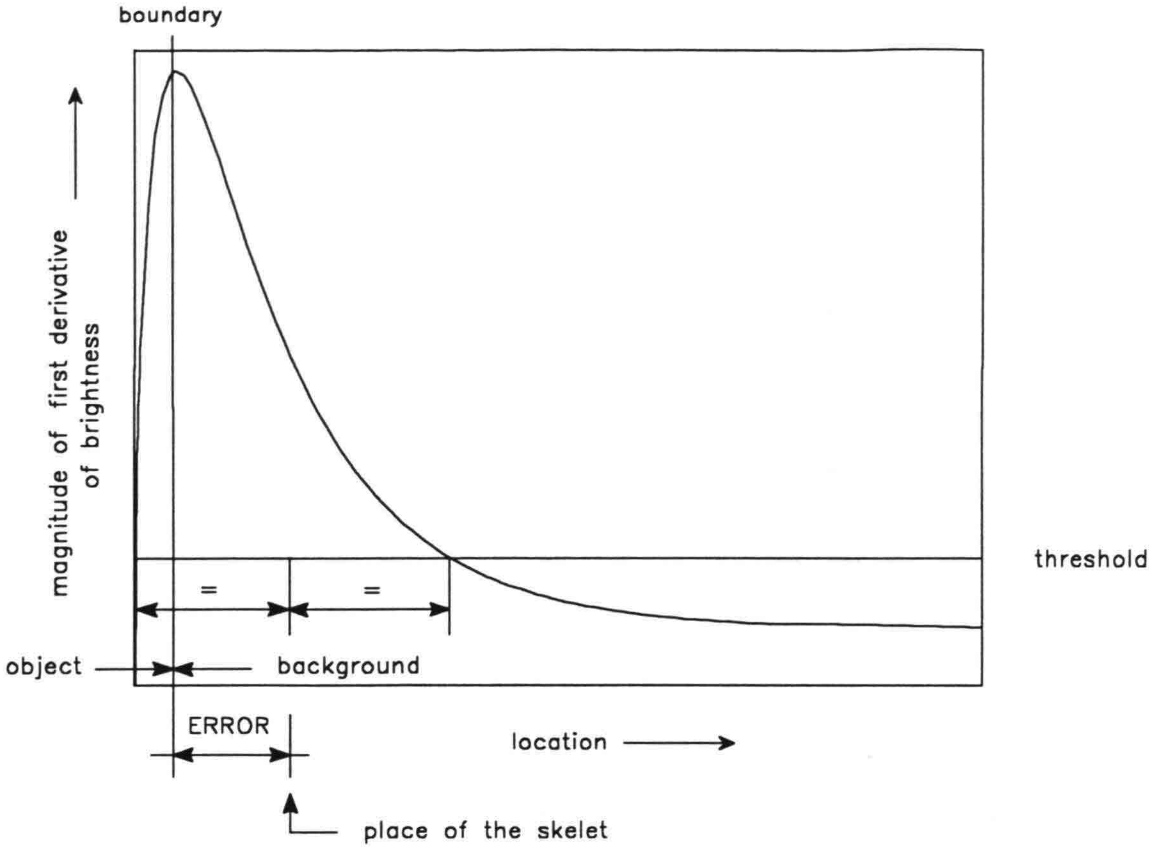


figure 3.4. In an asymmetric peak, the difference between top and skeleton causes an error if the boundary has been determined by skeletonation.

The error between the place of the skeleton and the place of the true boundary makes it impossible to correctly determine the boundaries by this method. On the top, the slope of the curve is equal to zero. This means that on that location the second derivative is equal to zero.

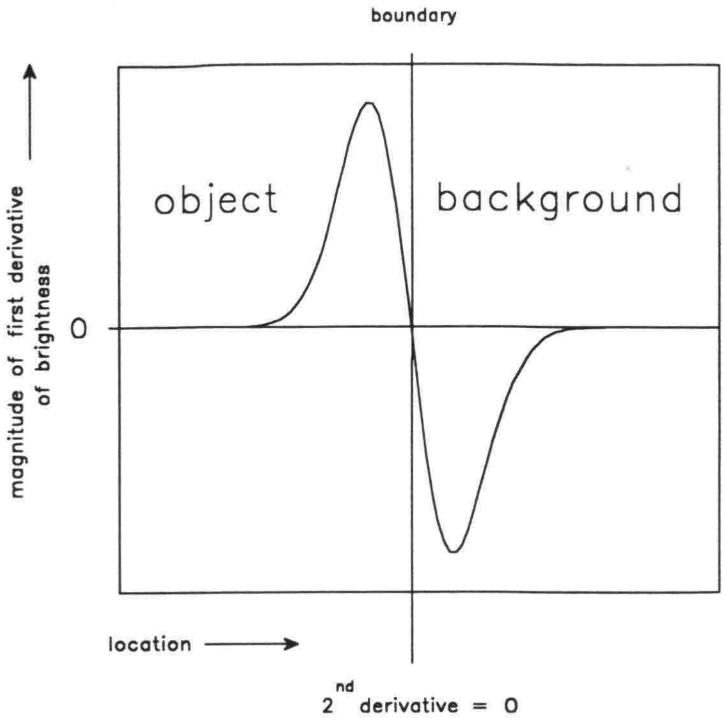


figure 3.5. The location of 2nd derivative=0 on the boundary between background and object.

Segmentation from the second derivative of the brightness

Second derivative image is determined by use of the laplace filter, the most important part of the second derivativ program. The property of this filter is, that when the objects are bright and the background dark, the second derivative inside the boundaries is greater than zero and outside little. At the boundaries the crossing is very sharp: from an extreme negativ to an extreme positiv value. This is also illustrated in figure below:

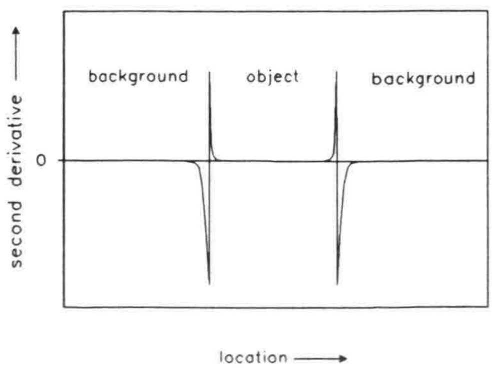


figure 3.6. property of laplace filter, objects bright; backgroud dark.

If the objects are dark and the background light (negative images), the inverse occurs:

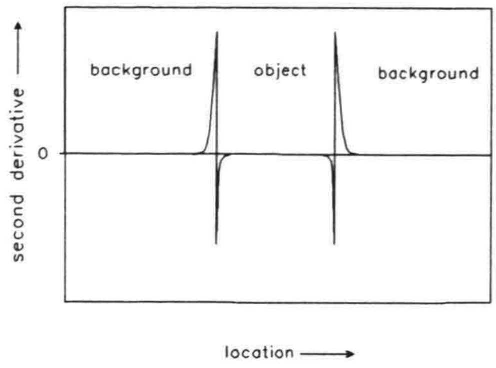


fig 3.7. property of laplace filter, objects:dark, background bright.

The process of distinguishing objects occurs by selecting areas where second derivative is greater than zero. In cases of large objects a possibility consists that in the middle the second derivative is equal to zero, so that the distinguished object has a shape of a ring. This problem can easily be solved by filling up the closed space.

There is still an another problem. Areas of second derivative greater than zero are not only from the objects but also from the fluctuations of the background noise, illustrated in figure 3.8:

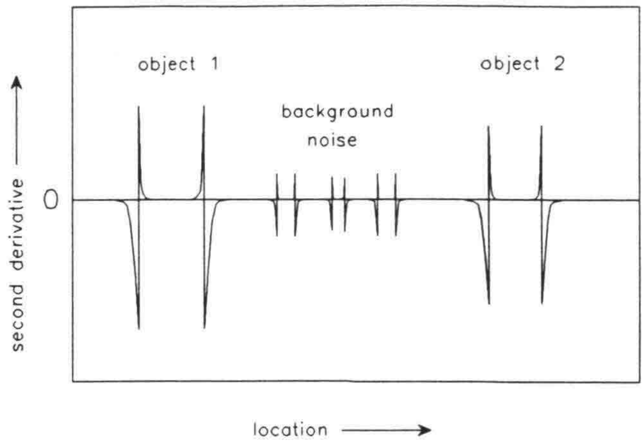


figure 3.8. Second derivative from two objects and background noise.

This can be resolved by selecting these areas with use of first derivative. Boundaries give peaks in the cross-section of the first derivative image peaks and segmentation with a threshold yields reference areas (fig. 3.9).

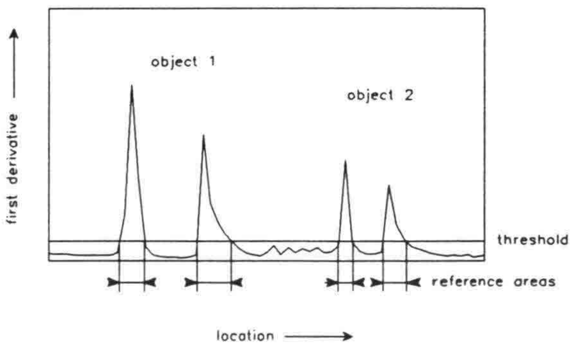


figure 3.9. Reference areas.

Boundaries of areas of second derivatives > 0 caused by the objects fall within reference areas, whereas those caused by the background noise do not, see fig. 3.10.

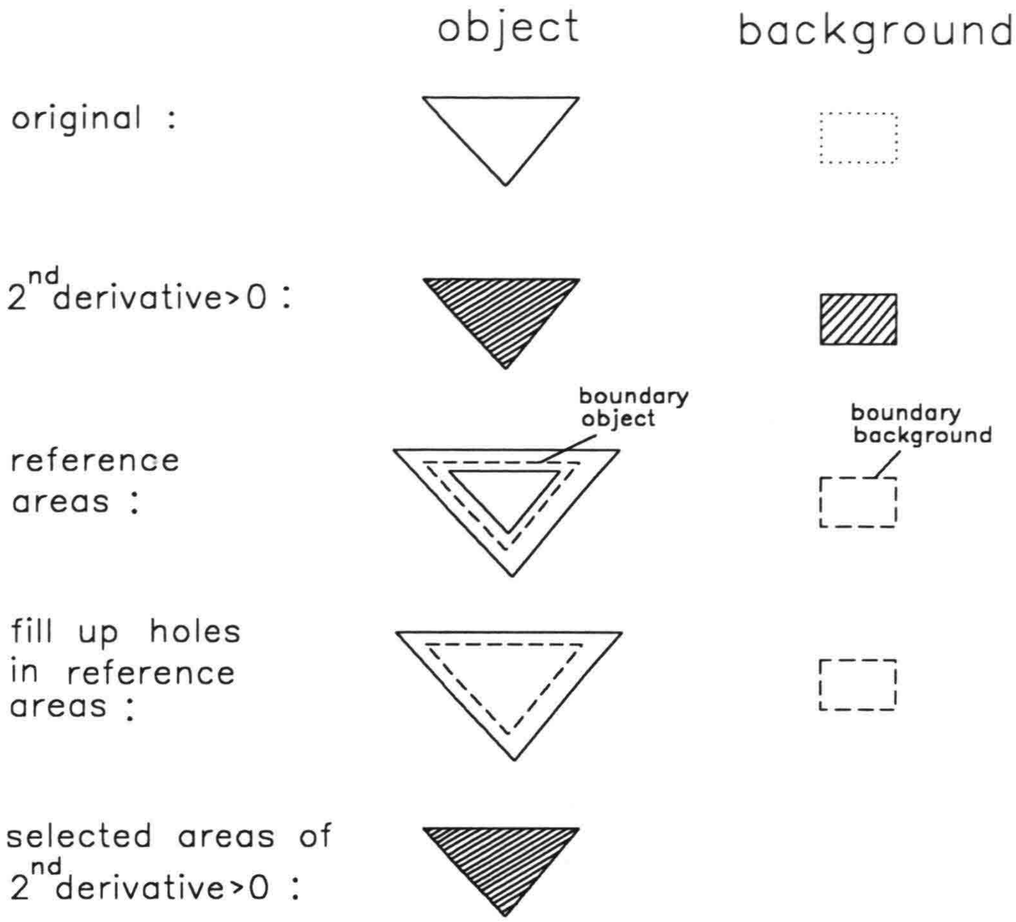


figure 3.10. The selection of the object from the background. In this illustration the object is imagined as a triangle and the background as a weak square box.

Segmentation of the reference areas is not possible with a constant threshold value for all images, because in some exposures the background fluctuates more than in others. As an example, an exposure with weak and another with strong background fluctuations (fig 3.11):

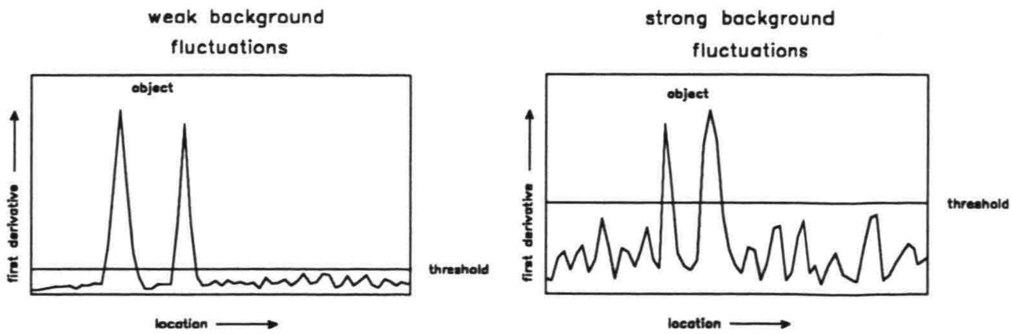


Figure 3.11. The dependence of the threshold on the background fluctuations.

The threshold depends on background fluctuations: the stronger the fluctuations the higher the threshold needed for segmentation.
The threshold is determined by use of the $n\sigma$ - method:

threshold = mean+$n\sigma$
mean: average value of background fluctuations;
σ : Root Mean Square of background fluctuations;
n : Real value (n>0).

The Root Mean Square is determined with:

$$\sigma = \sqrt{\overline{x^2} - \overline{x}^2}$$

with: x^2: the square of mean over all values, and x^2: mean over all square values
Details about this method and the way it has been implemented in the program are given in appendix C.1.1.

<u>*Segmentation direct from TCLi-package*</u>

Uniform filtering or minmax filtering followed by isodata thresholding.

In page 3 of this report it is explaned that direct thresholding is nearly impossible by presence of shading or background. Shading however can be removed with use of two kinds of filters: uniform filter and minmax filter. Direct threshold is determined iterativ by use of isodata threshold.

uniform filter

The original image is scanned with a moving window with given sizes. For each window position, pixel for pixel, the average value of the pixels within the window is calculated and stored into the pixel of the output which corresponds with the central pixel in the window. The size of the moving window is chosen such so that it is two times larger than the mean size of the objects. The result is an image that consists of only shading. So for correction shading can be removed by substraction with these images. The principle is also illustrated in figure below:

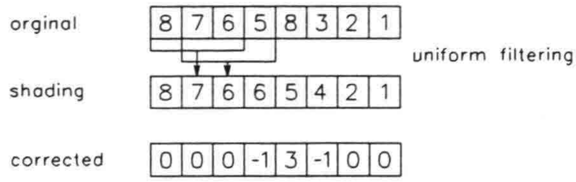


fig.: 3.10 principle of uniform filtering

minmax filter

Minmax filter can be split up into two parts:minimum filter and maximum filter, exactly said local minimum and local maximum filter.

Local minimum filter

Like uniform filter, the orginal image is scanned with a moving window with given sizes. For each window position pixel for pixel, the maximum value of the pixels within the window is determined and stored into the pixel of the output image which corresponds with the central pixel in the window.

Local maximum filter

The same as local minimum filter. Only in stead of the minimum value of the pixels within the moving window the maximum value is determined.
The result of the minmax filter is an image that consists of only shading. So for correction shading can be removed by substraction. The principle is also illustrated in figur below:

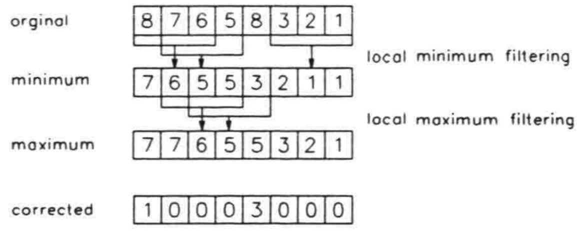


fig.: 3.11 principle of minmax filtering

isodata threshold

The isodata threshold is an iterative threshold based upon the grey value histogram of the image. The histogram is split up into two parts, the foreground and the background pixels. The mean value of the foreground and of the background pixels is calculated and a new threshold value is taken exactly between these two mean values. This process is repeated until the threshold value does not change anymore.

In comparison with the second derivative, in these methods is one disadvantange. Before analyzing the distribution the mean size has to be known for establish of the filter size.

3.2 Objects with holes

Cohesive sediment flocs are loose, fragile structures of clay and organic material, containing a lot of water. In some cases even holes are visible on the exposures. As the program determines equivalent object diameters based on object areas, the appropriate equivalent diameter of an object with a hole can only be obtained by including the surface of the hole. Therefore the program fills up the areas enclosed in objects.

3.3 Overlapping objects

Objects at the edge of the image and overlapping objects are only partly visible, and it is impossible to estimate their exact size. Therefore, the program detects all objects connected to the edge of the image and removes them before the size distribution is determined. The overlapping objects are not treated in a special way, because the envisaged use of the program does not include measuring high sediment concentrations. Consequently, overlapping of objects will rarely occur and not significantly influence the size distribution.

4. TESTS

The objectives of these test are: is the second derivative method useful for analyzing the size distribution and is it better than if one of the two other methods is used? It is expected that analysis with use of second derivative gives better results, if the cut off frequency of the background is low, owing its sensitivity of the noise fluctuations in high frequency-band, the property of second derivative.
To answer these questions is determined the deviations with regard to the reference, with other words the rightness.

For the brightness of the objects is taken two different situations:(1) uniform illumination; the brightness of the objects is equal everywhere, and (2) non-uniform illumination; there is a trend in the brightness of the objects, for example little dark at left and little bright at right.

Before to perform these tests original images is made first.

4.1 Creating original images

Objects

In the situation of uniform illumination two different values of maximum brightness of the objects are taken: 127 and 255 grey values (255=white). Maximum brightness, because the edges of the objects are not sharp and there is the brightness between 0 and maximum value (see fig.: 4.1). And in the situation of non-uniform illumination two different trends are taken: 127+0.125x (weak trend) and 127+0.250x (strong trend). At left the objects have a brightness of 127 and at right a brightness of 191 if the trend is weak or 255 if it is strong. (In further story of this report except for creating background behind objects (page 4) brightness of the objects means the maximum brightness)

The objects itself are created from a previous generated reference. The reference is an important subject of these tests. It defines which pixels belong to the objects or to the background and is showed in fig.: 3. The objects itself are created such that its bending-points lie exactly on the border between objects and background of the reference. The brightness on the places of the bending-points is exactly the half of the maximum brightness of the objects. It can be checked out with threshold of 127 for objects with a brightness of 255 and threshold of 63 for objects with a brightness of 127. The created objects are correct if the delivered binary images do not differ one pixel with the reference.

Background

The background is created by filtering of two dimensional white noise. Filtering occurs by convolution of this noise with the impulsrespons of the filter system. For convolution is used the theorem:

$$G(u,v)=F(u,v).H(u,v)$$

G(u,v) : fourier transform of the filtered image;
F(u,v) : fourier transform of image to be filtered: white noise;
H(u,v) : fourier transform of the impulsrespons;
u,v : co-ordinates in the fourier domain

For impulsrespons is used a two dimensional gauss curve of the form:

$$h(r)=C_1\exp\{-C_2|r|^2\}$$

C_1 : first constant, value = 1;
C_2 : second constant, used $1/r_c$;
$|r|$: length of the radius equal to $(x^2+y^2)^{½}$, Phytagoras.

So that the impulsrespons has the form:

$$h(x,y)=\exp\{-((x^2+y^2)^{½})^2/r_c)\}$$
$$h(x,y)=\exp\{-(x^2+y^2)/r_c\}$$

Background each with an another cut off frequency is created by justification of $r_{c.}$ If r_c is small, the broadness of the gausscurve is also small. The property of this curve is, that in the fourier domain it is also a gauss curve with a reciprocal broadness. Therefore in fourier domain the gauss curve is broad and after multiplication with the fourier transform of the white noise, the cut off frequency is high. When r_c is greater, the broadness of the gausscurve is greater and therefore in fourier domain it is smaller so that the cut off frequency of the filtered noise is smaller.

Creating background behind objects

Because the edges of the objects are not sharp, transmission takes place.

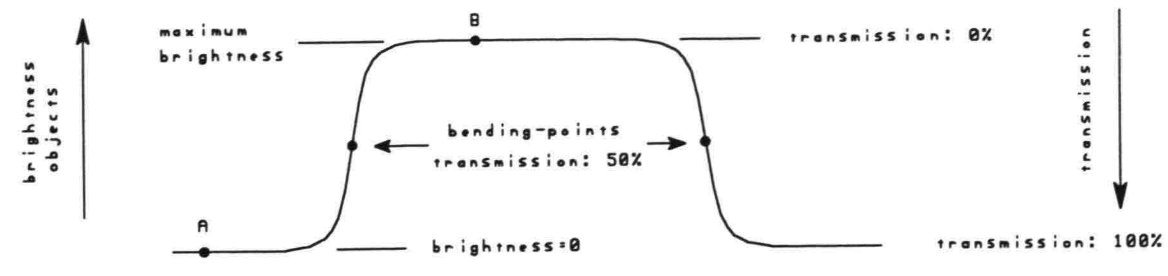


fig.: 4.1 Transmission through the edges of the object.

From background to object (from point A to B in above given figure) the transmission decreases from 100 to 0%. At bending-points the transmission is 50%.

For each pixel the transmission is calculated by use of

$$\text{transmission} = \frac{\text{maximum brightness - brightness object}}{\text{maximum brightness}}$$

The maximum brightness taken in these test are: for uniform illumination 127 or 255 and for non-uniform illumination 127+0.125x or 127+0.250x.

Pixel for pixel the background is added behind the object by use of

brightness of objects+transmission*brightness of background.

The simulated images are present in fig.: 2.1 to fig.: 2.4 (appendix) in which the cut off frequency of the background is 64.5 10^{-3} pixelunits^{-1} in the middle of high and low frequent noise. Fig.: 4 (appendix) shows the section plots of the simulated images, in which for example the brightness of the objects is 127 and the illumination is uniform. It presents the fluctuations of the background at three different cut off frequencies: high, middle and low.

5. RESULTS

The results of these tests are given in appendices: in the form of tables and in graphic (fig.: 5.1 until 5.4).

5.1 Interpretation

5.1.1 Second derivative

Uniform illumination

brightness objects: 255

The deviation of the average size with regard to the reference (Δmean) fluctuates between 0.08 and 0.14%, so round 0.1%, until cut off frequency is 300.10^{-3} pixelunits^{-1}. Above that frequency it increases about exponentially to 0.7% at 650.10^{-3} pixelunits^{-1}.

The deviation of the spread with regard to the reference (Δsigma or Δσ) lies round -0.25%: it fluctuates between -0.2 and -0.3%.

brightness objects: 127

Until 150.10^{-3} pixelunits^{-1} Δmean fluctuates between 0.02 and 0.07% so round 0.04%, so very low. Above 150.10^{-3} pixelunits^{-1} it increases to 0.3% at 300.10^{-3} pixelunits^{-1}. Δσ lies roughly on -0.05%: it fluctuates between -0.02 and -0.10% with a fluke of -0.15% at 8.10^{-3} pixelunits^{-1}. Above 100.10^{-3} pixelunits^{-1} it decreases to -0.35% at 300.10^{-3} pixelunits^{-1}.

If it is more than 300.10^{-3} pixelunits^{-1}, the influence of the background is too strong for good determination for objects with a brightness of 127. At 439.10^{-3} pixelunits^{-1} 73 objects are not detected and that is why Δmean and Δσ are -29 and -11% respectively. Above that frequency no particle is detected.

For objects with a brightness of 255 these difficulties are only present at no filtered white noise. Only 24 of 83 particles are not detected so that Δsigma and Δσ are -7 and -6% respectively.

non-uniform illumination

Both for weak and for strong trend in illumination, until 200.10^{-3} pixelunits^{-1} Δmean fluctuates round 0.15%. Then from 200.10^{-3} pixelunits^{-1} it increases to 1.4% at 450.10^{-3} pixelunits^{-1} if the trend is weak and to 0.55% at 650.10^{-3} pixelunits^{-1} if it is strongly. For both trends Δσ fluctuates round -0.25% until 300.10^{-3} pixelunits^{-1}. Then it begins to fluctuate strong.

Until 200.10^{-3} pixelunits^{-1} the deviations in mean and sigma are very low and are constant round 0.15% for the mean and -0.25% for the sigma. From that frequency the influence of the background is increasing clearly and from more than 450.10^{-3} pixelunits^{-1} good determination is not possible any more because not all objects are detected: for example at weak trend at 645.10^{-3} pixelunits^{-1} 46 of 83 objects are detected so that Δmean and Δσ are -2.8 and -6.2% respectively.

5.1.2 Uniform filter + isodata threshold

Uniform illumination

brightness objects: 255

The values of Δmean increase first from -1.1% at 8.10^{-3} pixelunits^{-1} to about -0.66% at 150.10^{-3} pixelunits^{-1}, then decrease to -1.1% at 645.10^{-3} pixelunits^{-1}. Δσ fluctuates between -6.6 and -7.0%, so round -6.8% The threshold lies between 125 and 135 and from 195.10^{-3} pixelunits^{-1} it shows an increasing trend.

brightness objects: 127

Until 150.10^{-3} pixelunits^{-1} Δmean shows an increasing trend from -1.2 to -0.5%., then decreases to rounded off -25% at 645.10^{-3} pixelunits^{-1}.

Δσ fluctuates between -6.8 and -8.2% until 150.10^{-3} pixelunits^{-1}. If the cut off frequency is more than 150.10^{-3}, Δσ increases while Δmean decreases. In spite of a little bit higher value than at lower cut off frequency, the threshold is low enough to segmentate behind objects also tops of the background noise. It detects more objects than of the reference. In the bitplane these are recognized as very small areas of about one or 5 pixels. This occurs when it is 195.10^{-3} pixelunits^{-1} or more.

non-uniform illumination

weak trend

The values of Δmean lie between -1.1 and -1.7% and of Δσ between -7 and -7.5%. The threshold increases with the cut off frequency: from 84 at 8.10^{-3} pixelunits^{-1} to 108 at 645.10^{-3} pixelunits^{-1}.

strong trend

With regard to the weak trend the rightness in the mean values is less while that of the sigma is the same. Δmean lies between -1.6 and -2.0%. For Δσ the values lie between -7.0 and -7.6%. The threshold increases with the cut off frequency: from 108 at 8.10^{-3} pixelunits^{-1} to 121 at 645.10^{-3} pixelunits^{-1}.

5.1.3 Minmax filter + isodata threshold

Uniform illumination

brightness objects: 255

The values of Δmean and Δσ lie between 0.1 and 0.3%, so for both constant round

0.2%.

brightness objects: 127

The rightness is less than when the objects have a clearness of 255. Except at 8.10^{-3} pixelunits^{-1} Δmean gives a little trend from 0.3% at 10.10^{-3} pixelunits^{-1} to 0.55% at 650.10^{-3} pixelunits^{-1}. An extreme value of -0.1% lies on 8.10^{-3} pixelunits^{-1}. Until 20.10^{-3} pixelunits^{-1} $\Delta\sigma$ decreases rapidly from 1.6% at 8.10^{-3} pixelunits^{-1} to -0.1%, then increases to 0.8% at 650.10^{-3} pixelunits^{-1}, with a steady state at 0.4% between 80 and 500.10^{-3} pixelunits^{-1}.

non-uniform illumination

weak trend

Δmean shows an increasing trend from -0.7% to 0.0% until 30.10^{-3} pixelunits^{-1} and then fluctuates between -0.2 and 0.2%, so round 0.0%. The course of $\Delta\sigma$ is probably inverse of that of Δmean. The first part of figure 5.4 seems to show a decreasing trend from 2.8% at 10.10^{-3} pixelunits^{-1} to 1.5% at 30.10^{-3} pixelunits^{-1}, and the next part seems to be constant round 1.5%.

strong trend

Until 100.10^{-3} pixelunits^{-1} Δmean shows an increasing trend from -1.9 to about -1.0% and then constant round that value. This shape is like the one of the weak trend, only seen in absolute values it is greater. The values of $\Delta\sigma$ shows also the same course as that of the weak trend. However they are greater: until 100.10^{-3} pixelunits^{-1} they decrease from 4.5 to 3.2%, and then constant between 2.7 and 3.0%, so round 3.2%.

5.2 Comparison of the three methods

The second derivative gives the best results in the situation of non uniform illumination. Until cut off frequency of 300.10^{-3} pixelunits^{-1} the values of Δmean and $\Delta\sigma$ are 0.15% and -0.25% respectively. Above that frequency the influence of the background noise begins to play an important roll.

However if the illumination is uniform, the results of this method is nearly the same as that of the minmax. Also in that situation the sensitivity of the background play an important roll if the cut off frequency is more than 200 or 300.10^{-3} pixelunits^{-1}. The rightness of uniform method is in both situations the most less of the three.

non-uniform illumination

mean

If the average size of the objects is determined and there is a weak trend in the illumination, there is no difference in the rightness if the second derivative or the minmax method is used, except at low cut off frequency. In a range until 15.10^{-3} pixelunits^{-1} the second derivative is better, from 15.10^{-3} until 300.10^{-3} pixelunits^{-1} the rightness of each other are equal and above that frequency is it of the minmax better.

If the trend is stronger (127+0.25x) the rightness of the minmax decreases while that of the second derivative is constant. Until 300.10^{-3} pixelunits^{-1} the rightness of the second derivative is much better, so about 5 or 10 times.

In comparison with the minmax and the second derivative, the rightness of the uniform filter method is less. The deviation lies between -1.2 and -1.6% against -0.6 and 0.2% for the minmax if the trend is weak. Is the trend stronger the rightness is still less: so about -1.6 to -2.0%.

sigma

For determination of the sigma up to about 300.10^{-3} pixelunits^{-1} it is clear that the rightness of the second derivative is the best. Comparison with the minmax, for the weak and strong trend the rightnesses are about 6 and 15 times better respectively.

Usage with the uniform filter gives for both trends a deviation of about -7 or -7.5%; it seems independent on the strongness of the trend. Because the deviation of the second derivative is round -0.25% the rightness of the uniform filter method is 25 or 30 times less.

Uniform illumination

With regard to the uniform filter method, the result of the second derivative method are clearly better. However with minmax method the results of both are nearly the same. Only in determination of the mean, if the objects have a brightness of 127 and if the cut off frequency is below 200 or 300.10^{-3} pixelunits^{-1},is that of the second derivative better. And finally above that frequency the rightness of the minmax method is the best. For determination of the spread there is no differ in results.

mean

Below 200.10^{-3} pixelunits^{-1} and if the objects have a brightness of 127, Δmean fluctuates round 0.05%, while that of the minmax method with exception of a measurement at very low cut off frequency round 0.5%, and of the uniform method mainly round -1.0%.

If the brightness is 255 and the cut off frequency below 300.10^{-3} pixelunits^{-1} , Δmean fluctuates round 0.1%, while that of the minmax method round 0.2% and of the uniform method round -1.0%.

Therefore with regard to minmax and uniform method until 200 or 300.10^{-3} pixelunits^{-1}, the second derivative gives respectively about 2 and 10 times more rightness if the brightness is 255 and respectively about 10 and 20 times or more if the brightness is 127 grey values.

sigma

For objects with a brightness of 127 or 255, the rightness of the second derivative or of the minmax method are the same. Only a few places in the cut off frequency band, the second derivative is better: until 15.10^{-3} pixelunits^{-1} and in the range of 80 until

250.10^{-3} pixelunits^{-1}, when the brightness is 127.
It seems that if the brightness is 255, the minmax method gives somewhat better results. Seen in absolute values $\Delta\sigma$ fluctuates round 0.15 or 0.2% against 0.25%.

With regard to uniform filter method the second derivative is much better. At both brightness of the objects $\Delta\sigma$ is about -7%, except in the range above 150.10^{-3} pixelunits^{-1} when the brightness is 127. It means that second derivative is 30 times better if the brightness is 255 and 60 if it is 127 and below 200.10^{-3} pixelunits^{-1}.

6. CONCLUSIONS

From these tests the following conclusions can be drawn:

Second derivative method

Generally with regard to the true values and if the cut off frequency of the background noise is below 200 or 300.10^{-3} pixelunits^{-1}, the deviation in the calculated mean size and the spread with use of second derivative method seen in absolute values(*) is below 0.25%. In the situation of uniform illumination are these for the mean and the spread respectively 0.05 and 0.15% or below if the objects have a brightness of 127 grey values and respectively 0.1 and 0.2% if they have a brightness of 255. Above that frequency the background noise begins to play a too important roll for good determination.

In the situation of non-uniform illumination, the rightness is not depended on the strongness of the trend in the brightness of the objects. The deviations in the calculated mean and the sigma are 0.15 and 0.25% respectively.

Minmax method

In the situation of uniform illumination, the deviations in the calculated mean and the sigma are both round 0.2% if the brightness of the objects is 255 and 0.5% if it is 127.

In the situation of non-uniform illumination: (1) if the trend is weak, the deviations for determination of the mean and the sigma are respectively 0.25% or below if the cut off frequency is not very low and 2%; (2) if it is strong they are respectively 1.2 and 4%. This means that the rightness decreases with the trend in the brightness of the objects.

Uniform method

In the situation of uniform illumination, the deviations in the calculated mean and the sigma are round 1 and 7% respectively if the objects have a brightness of 255, and respectively between 0.5 - 1.5% and round 7% if they have a brightness of 127.

In the situation of non-uniform illumination, these deviations are respectively between 1 - 1.5% and 7 - 7.5% if the trend is weak and respectively 1.5 - 2% and 7 - 7.5% if it is strong. It seems that the rightness for determination of the mean decreases with the strongness of the trend in illumination, and not changes for the sigma.

(*) **The percentage descripted in these conclusions are absolute values**

Comparisons with the minmax and uniform method

In the situation of uniform illumination the second derivative is not appreciable better than the minmax method. The rightness of both are nearly equal if the objects are bright. The rightness of the second derivative is be better and that of the minmax method less if the brightness of objects is decreased until 127 grey values.

The rightness of the second derivative is clearly better if the illumination is not uniform, except if the trend is weak and only the average size is determined. In that situation the rightness of both are nearly the same.

With regard to the uniform filter method, in the situations at all the rightness of the second derivative is clearly better.

Well, these comparisons are only valid until 200 or 300.10^{-3} pixelunits^{-1}, because its sensitivity of the background noise in high frequency band.

7. LITERATURE

[1] - A Kelly and I Pohl,
An introduction to programming in C
The Benjamin/Cummings Publishing Company,Inc (1984)
TCL-Image User's Manual, Part Two

[2] - TU Delft, UvA, TPD
Image Processing for Industrial Applications,
Introductory Course: 28-29 Oktober 1991

[3] - manuals of Multihouse TSI:
TCL-Image User's Manual, Part One
TCL-Image User's Manual, Part Two
TCL-Image Programmer's Manual

APPENDIX A

FIGURES

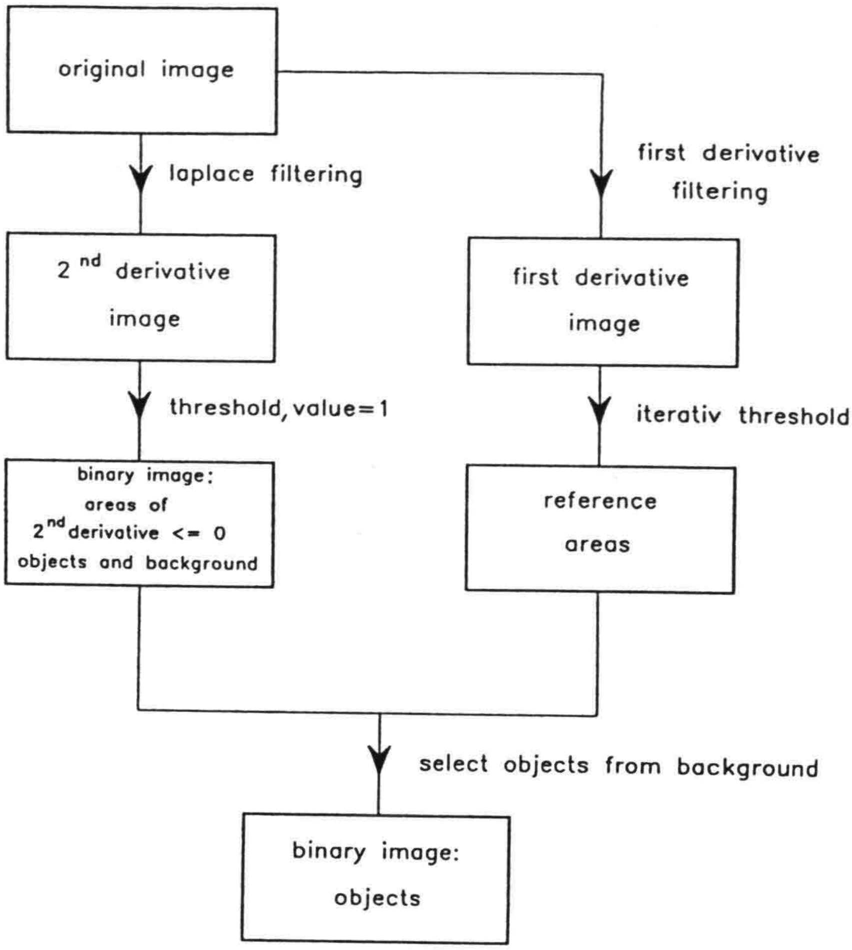


FIG. 1.1

SECOND DERIVATIVE

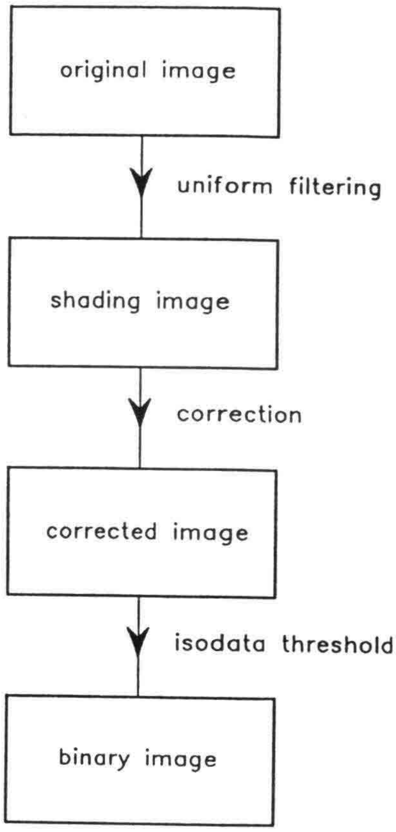


FIG. 1.2

UNIFORM FILTER
+
ISODATA THRESHOLD

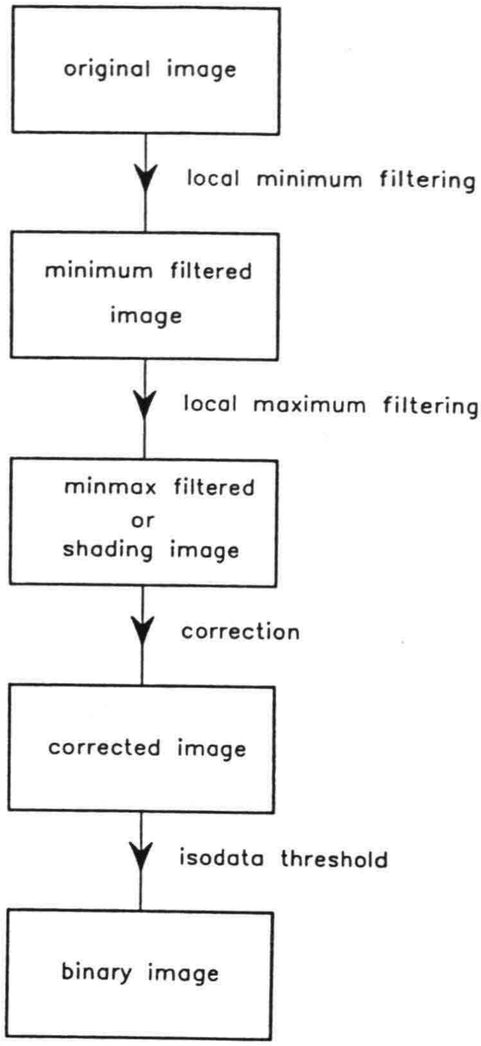


FIG. 1.3	MINMAX FILTER + ISODATA THRESHOLD

brightness: 255

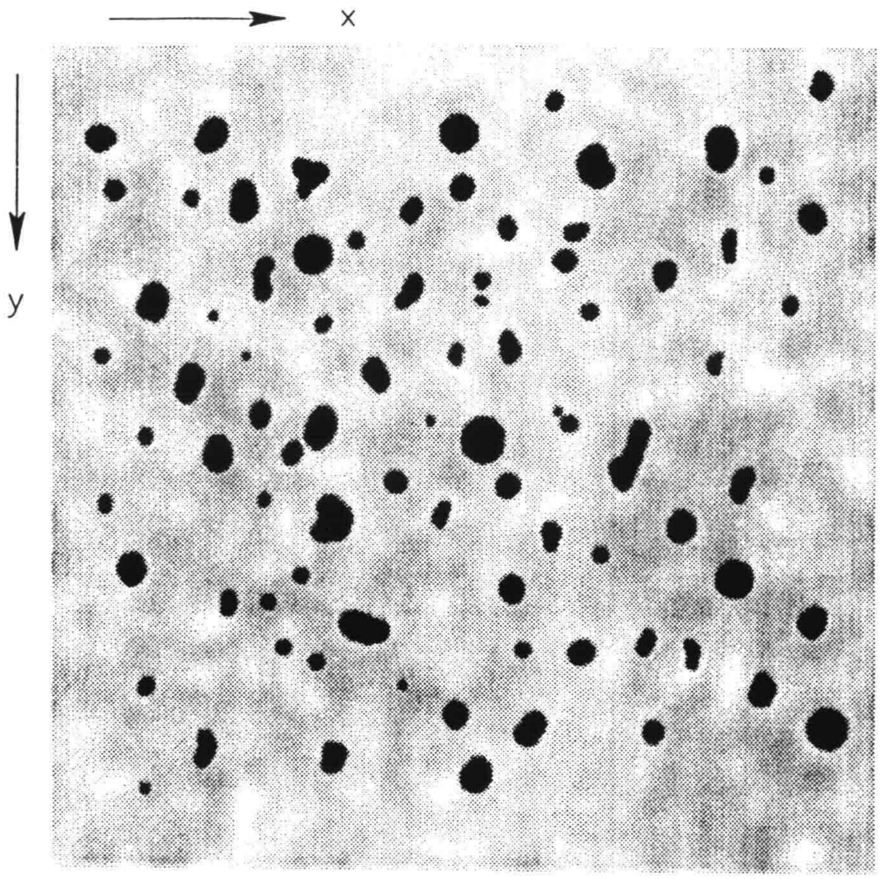


FIG. 2.1	SIMULATED IMAGE constant brightness objects

brightness: 127

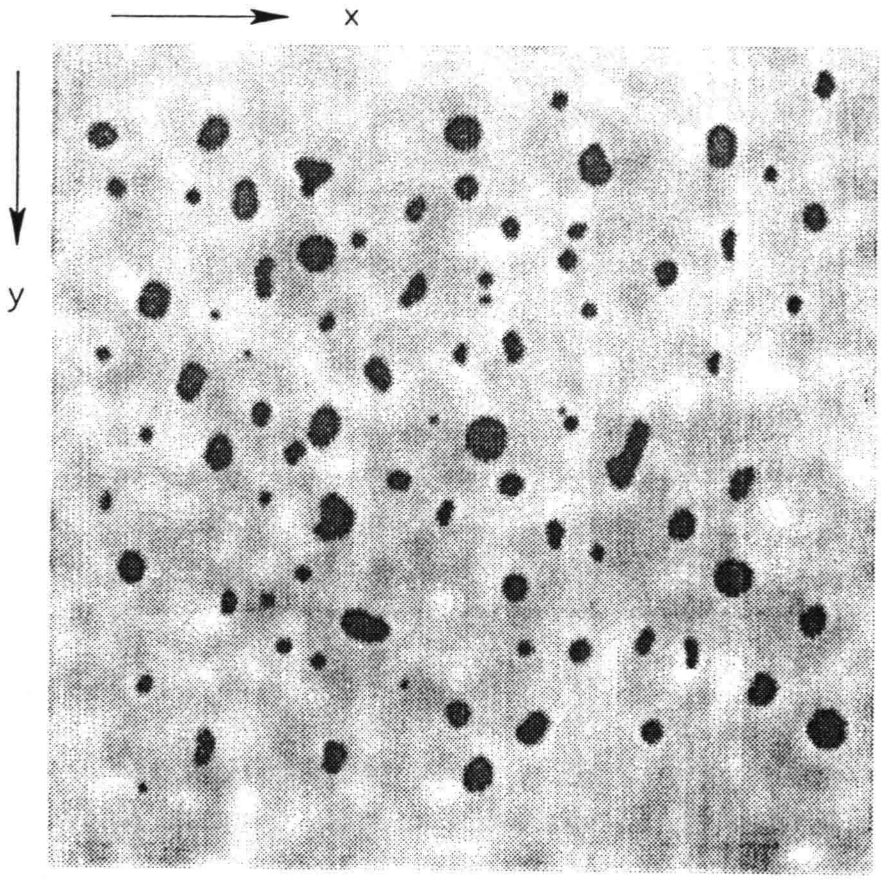


FIG. 2.2	SIMULATED IMAGE **constant brightness objects**

brightness: 127+0.125x

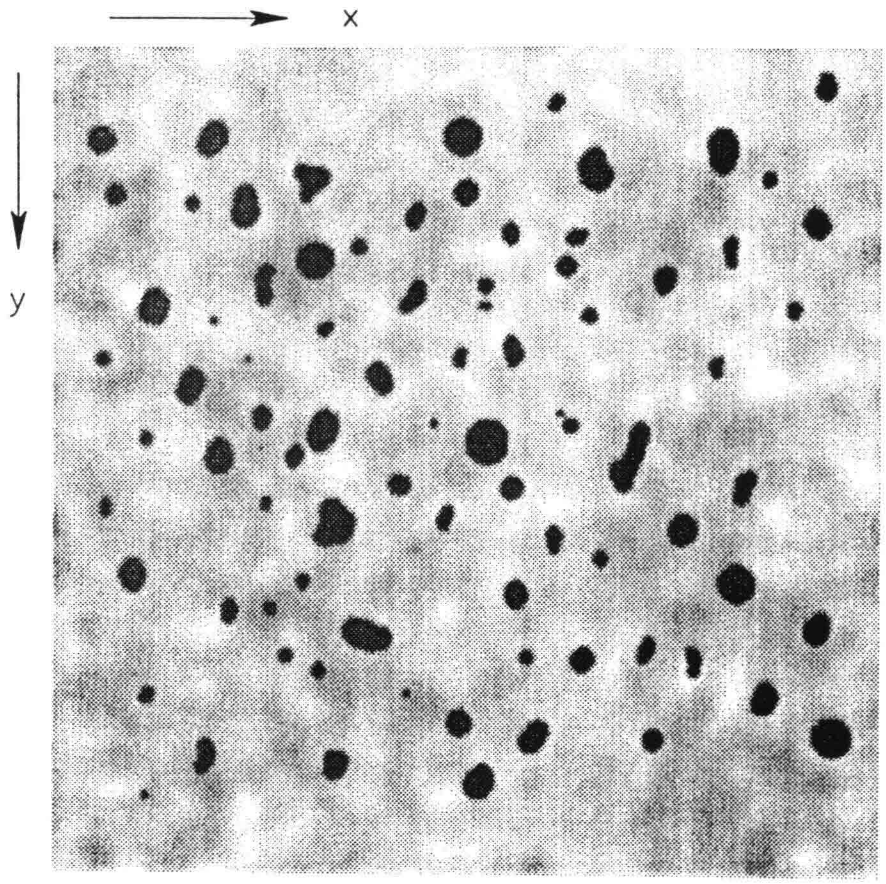


FIG. 2.3	SIMULATED IMAGE trend in brightness objects

brightness: 127+0.250x

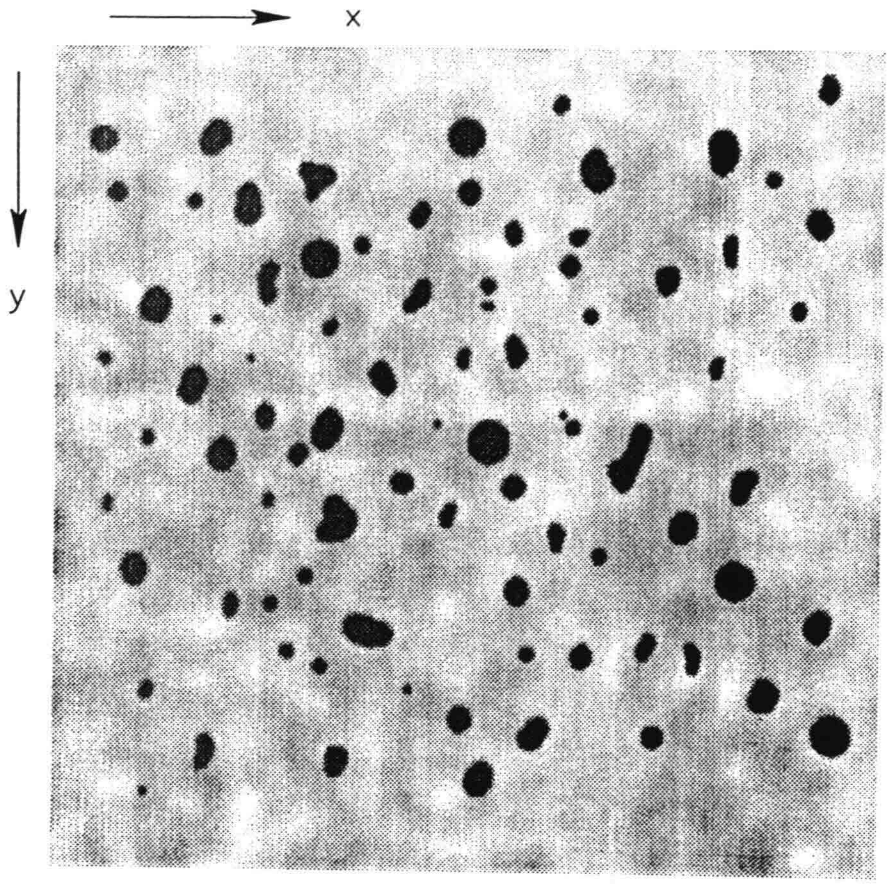


FIG. 2.4	SIMULATED IMAGE trend in brightness objects

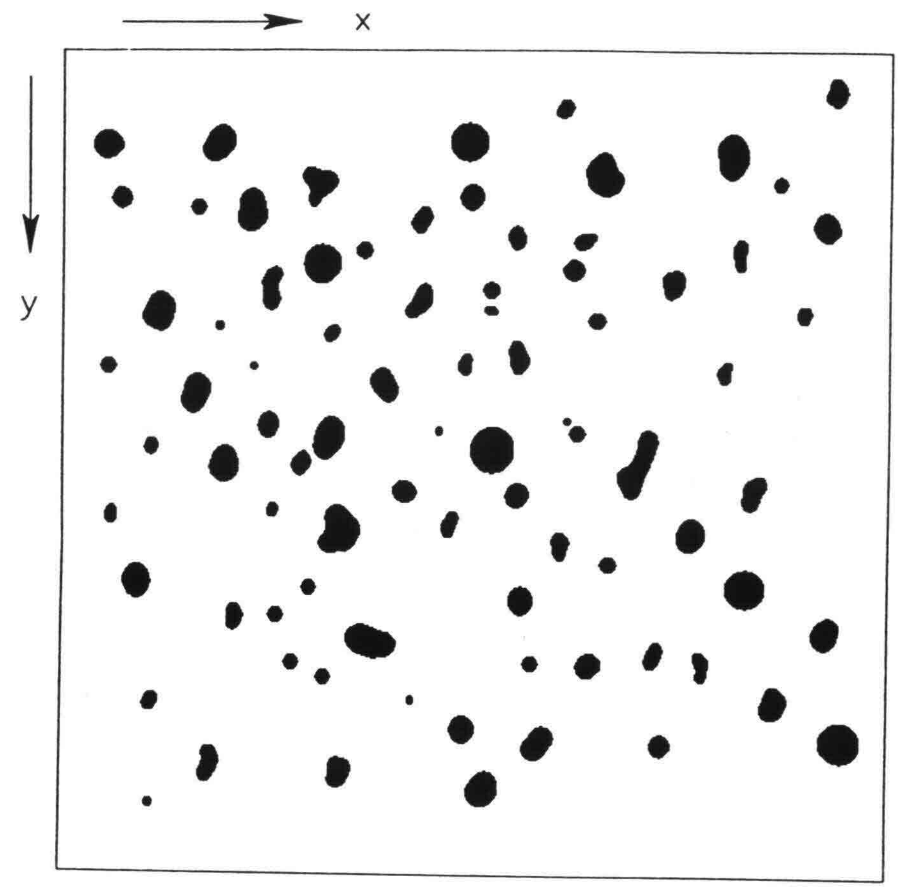


FIG. 3

REFERENCE

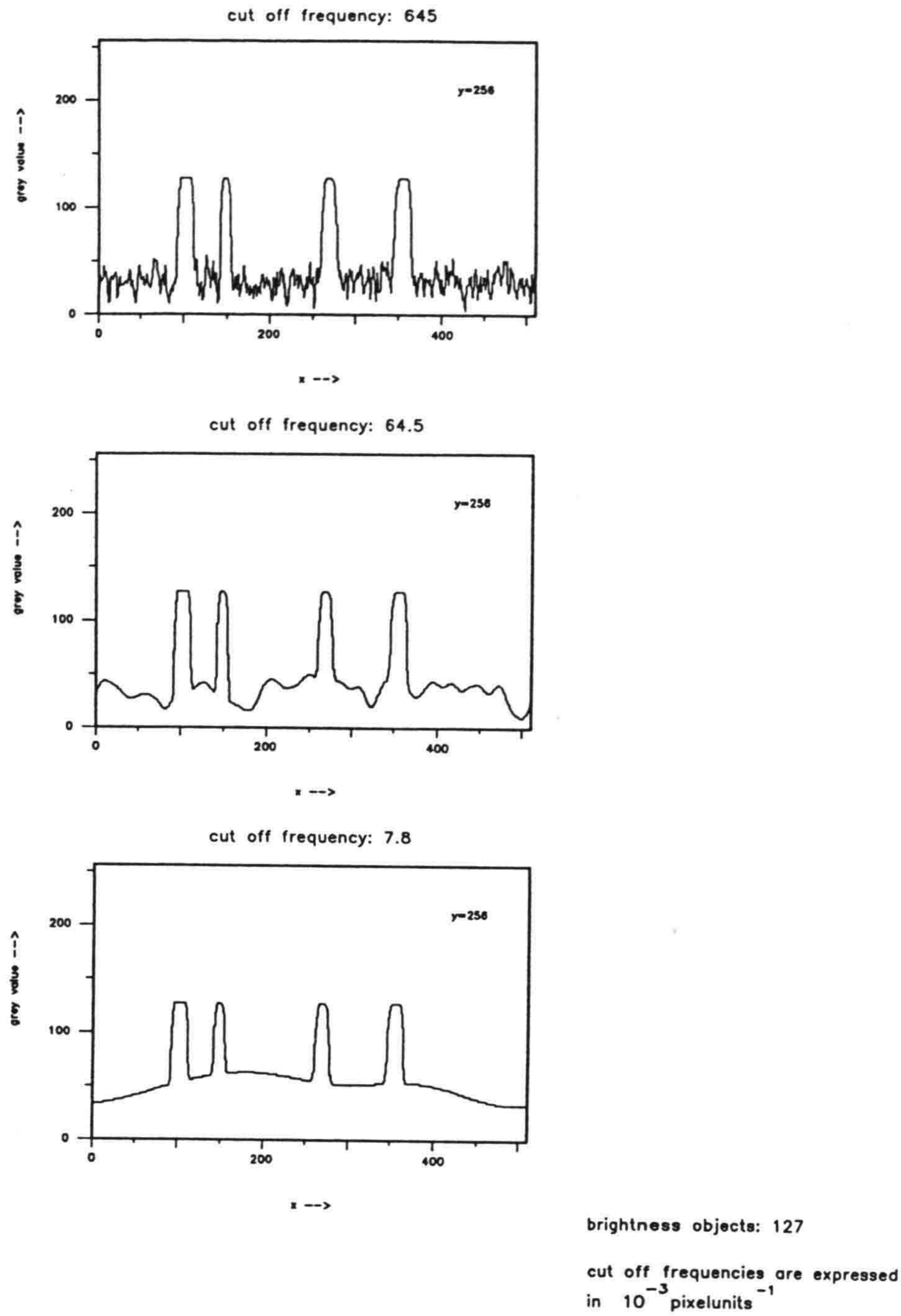
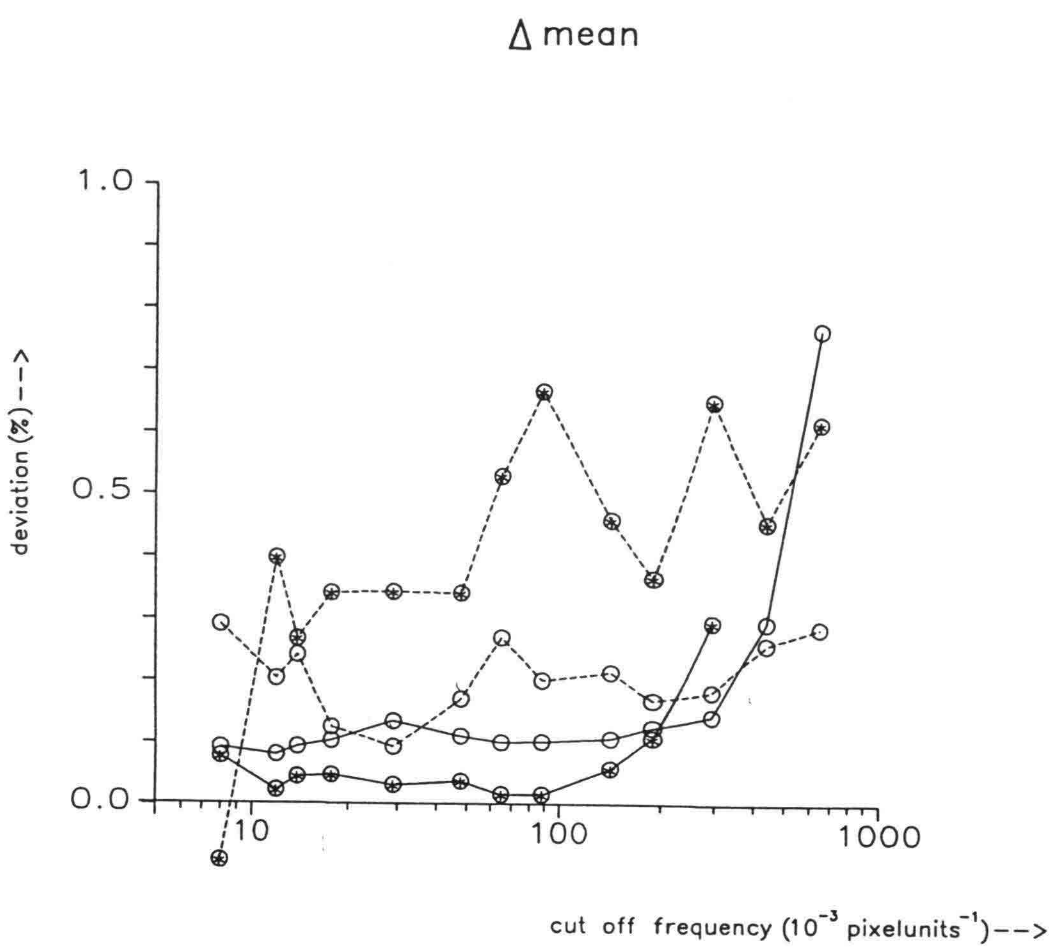


FIG. 4	SECTION PLOTS OF OBJECTS WITH BACKGROUND three different cut off frequencies



explanation lines and symbols:

lines:

——— 2^{nd} derivative

- - - - - - minmax + isodata

symbols:

⊛ brightness: 127

O brightness: 255

FIG. 5.1	CONSTANT BRIGHTNESS OBJECTS

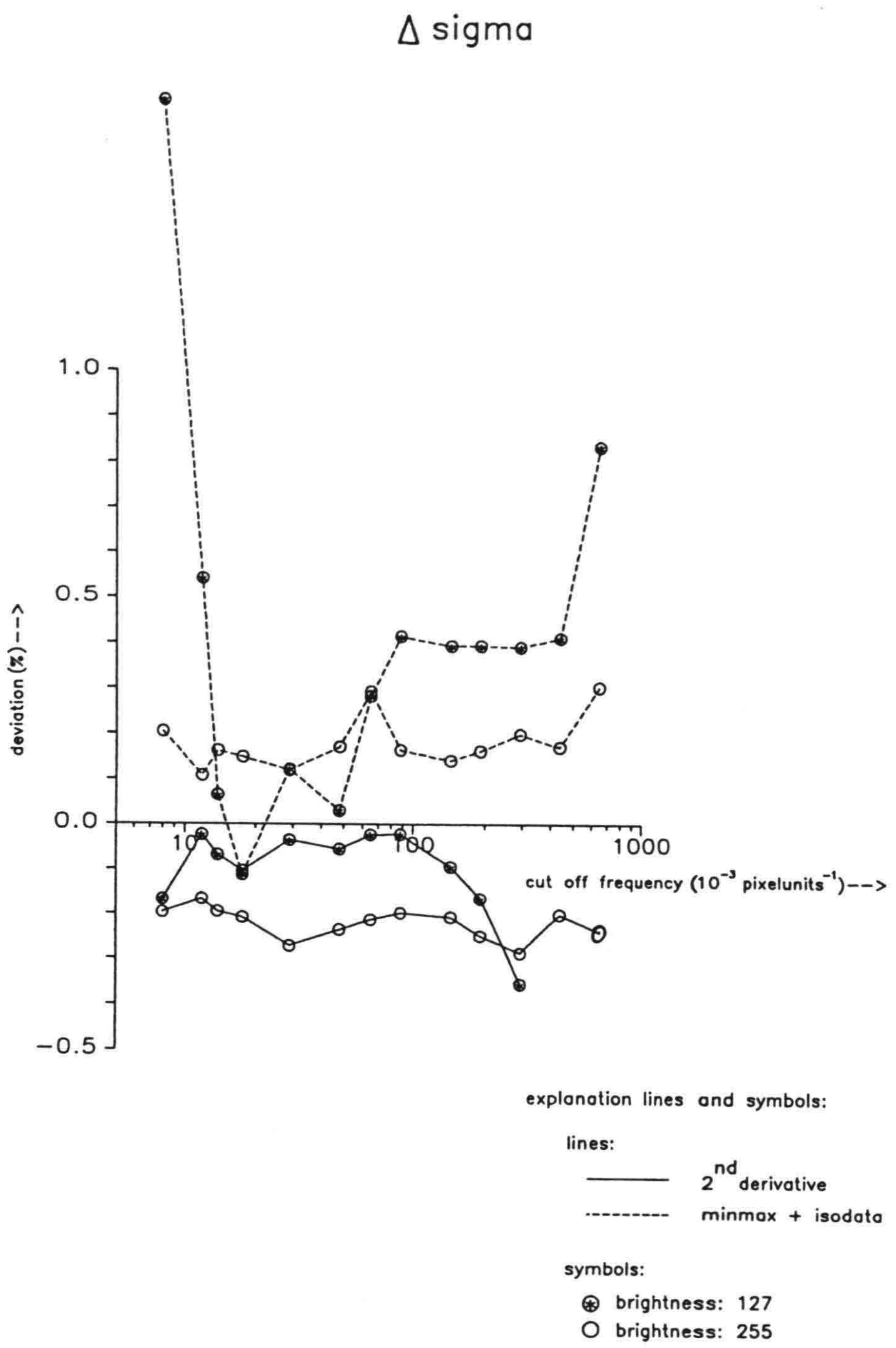


FIG. 5.2

CONSTANT BRIGHTNESS OBJECTS

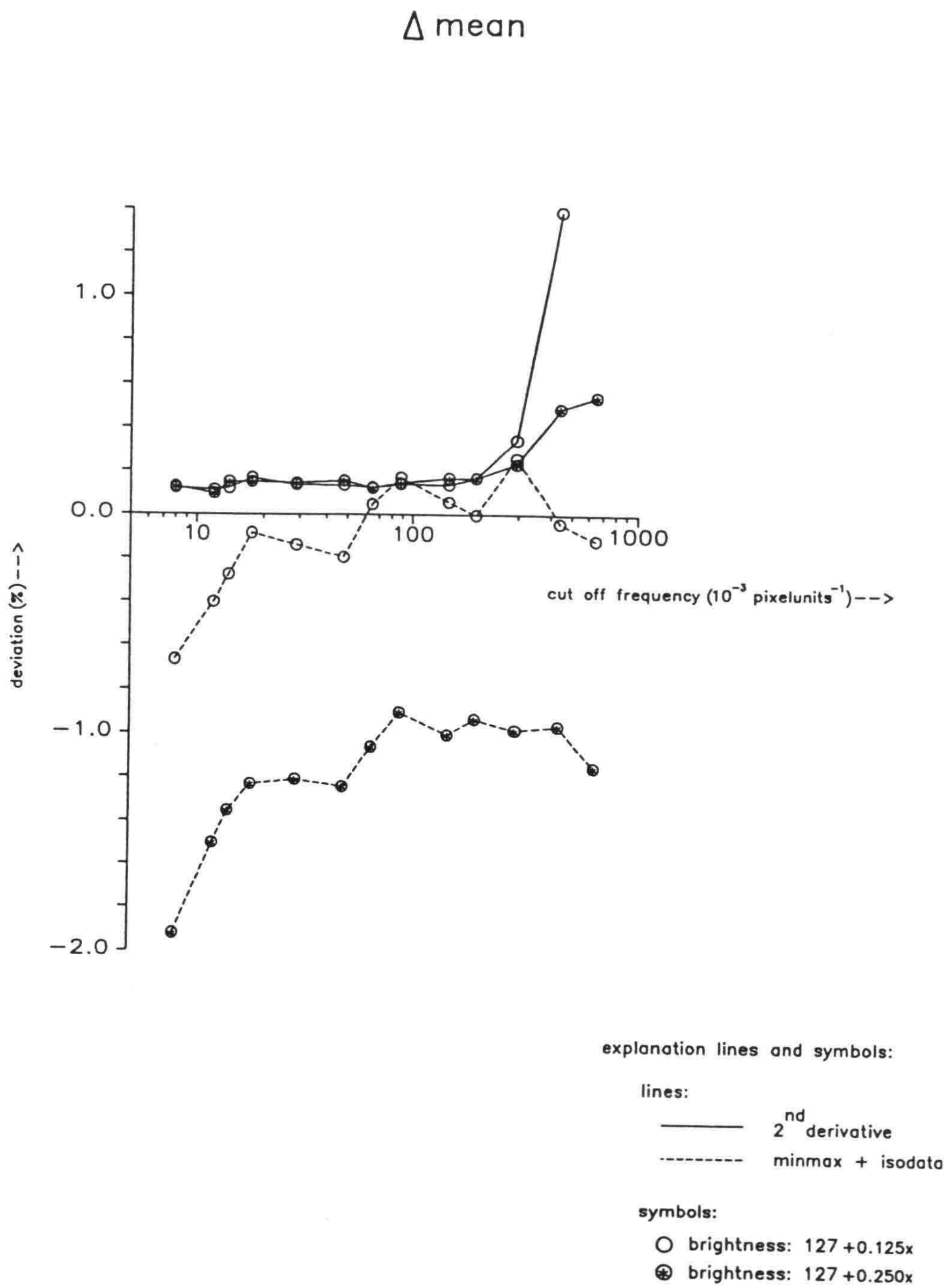


FIG. 5.3

TREND IN BRIGHTNESS OBJECTS

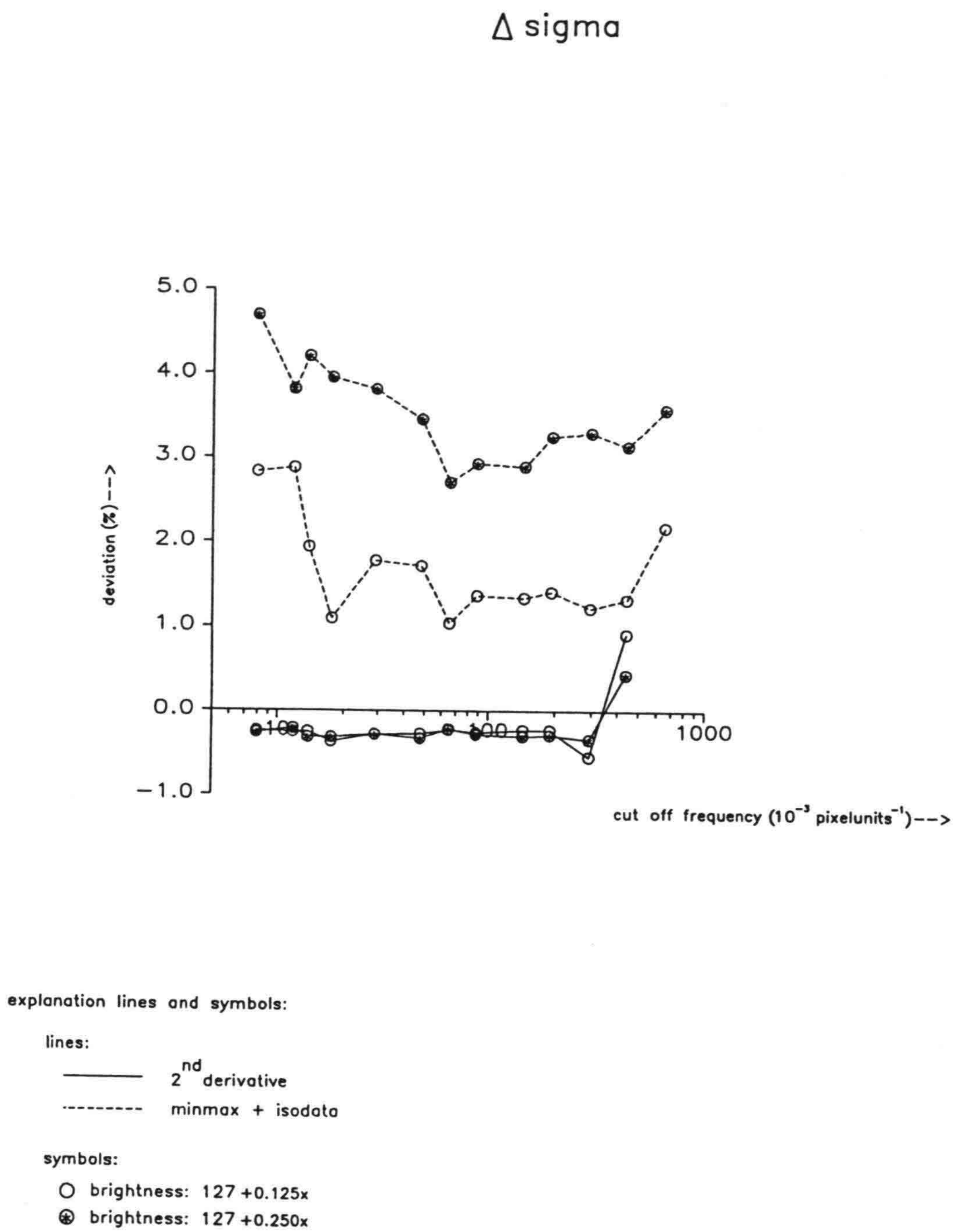


FIG. 5.4

TREND IN BRIGHTNESS OBJECTS

APPENDIX B

BRIGHTNESS

SECOND DERIVATIVE

Brightness objects: constant

Brightness: 127

cut off frequency (10^{-3} pixelunits^{-1})	N	mean (pixelunits)	σ (pixelunits)	ΔN	Δmean (%)	Δσ (%)
8	83	15.432	5.710	0	0.078	-0.168
12	83	15.424	5.718	0	0.023	-0.024
14	83	15.427	5.715	0	0.045	-0.069
18	83	15.427	5.713	0	0.047	-0.104
29	83	15.425	5.717	0	0.031	-0.036
48	83	15.426	5.716	0	0.037	-0.056
65	83	15.423	5.718	0	0.016	-0.024
88	83	15.423	5.718	0	0.016	-0.023
146	83	15.429	5.714	0	0.058	-0.095
195	83	15.437	5.710	0	0.107	-0.166
293	83	15.446	5.699	0	0.293	-0.355
439	83	10.916	5.090	-73	-29.213	-11.012
645	83	------	-----	-83	-----	-----

Brightness: 255

cut off frequency (10^{-3} pixelunits^{-1})	N	mean (pixelunits)	σ (pixelunits)	ΔN	Δmean (%)	Δσ (%)
8	83	15.434	5.708	0	0.092	-0.197
12	83	15.433	5.710	0	0.081	-0.167
14	83	15.435	5.708	0	0.094	-0.195
18	83	15.436	5.707	0	0.103	-0.208
29	83	15.441	5.704	0	0.134	-0.272
48	83	15.437	5.706	0	0.110	-0.235
65	83	15.436	5.707	0	0.100	-0.213
88	83	15.436	5.708	0	0.101	-0.198
146	83	15.437	5.707	0	0.106	-0.207
195	83	15.439	5.705	0	0.124	-0.249
293	83	15.442	5.703	0	0.142	-0.287
439	83	15.465	5.708	0	0.292	-0.201
645	83	15.532	5.706	0	0.762	-0.239

SECOND DERIVATIVE

Brightness objects: trend

Brightness: 127+0.125x

cut off frequency (10^{-3} pixelunits^{-1})	N	mean (pixelunits)	σ (pixelunits)	ΔN	Δmean (%)	Δσ (%)
8	83	15.439	5.706	0	0.120	-0.241
12	83	15.438	5.705	0	0.112	-0.252
14	83	15.439	5.705	0	0.119	-0.252
18	83	15.446	5.699	0	0.169	-0.364
29	83	15.441	5.703	0	0.135	-0.281
48	83	15.441	5.704	0	0.134	-0.270
65	83	15.440	5.706	0	0.125	-0.227
88	83	15.442	5.705	0	0.138	-0.254
146	83	15.442	5.706	0	0.137	-0.227
195	83	15.446	5.706	0	0.166	-0.226
293	83	15.473	5.688	0	0.341	-0.539
439	76	15.634	5.772	-7	1.385	0.913
645	46	15.986	5.366	-37	-2.817	-6.179

Brightness: 127+0.250x

cut off frequency (10^{-3} pixelunits^{-1})	N	mean (pixelunits)	σ (pixelunits)	ΔN	Δmean (%)	Δσ (%)
8	83	15.440	5.704	0	0.126	-0.260
12	83	15.435	5.707	0	0.096	-0.209
14	83	15.443	5.701	0	0.149	-0.313
18	83	15.443	5.701	0	0.149	-0.313
29	83	15.443	5.703	0	0.145	-0.281
48	83	15.444	5.701	0	0.154	-0.324
65	83	15.439	5.707	0	0.120	-0.215
88	83	15.443	5.703	0	0.144	-0.278
146	83	15.446	5.702	0	0.167	-0.301
195	83	15.446	5.703	0	0.165	-0.285
293	83	15.456	5.700	0	0.230	-0.336
439	80	15.495	5.744	-3	0.483	0.435
645	70	15.503	5.802	-13	0.538	1.453

MINMAX FILTER + ISODATA THRESHOLD

Brightness objects: constant
Size of filter: 31

Brightness: 127

cut off frequency (10^{-3} pixelunits^{-1})	N	mean (pixelunits)	σ (pixelunits)	ΔN	Δmean (%)	Δσ (%)	threshold value
8	83	15.406	5.811	0	-0.090	1.600	37
12	83	15.482	5.750	0	0.398	0.541	38
14	83	15.462	5.723	0	0.268	0.064	42
18	83	15.473	5.713	0	0.341	-0.113	44
29	83	15.470	5.726	0	0.342	0.121	45
48	83	15.473	5.721	0	0.340	0.029	46
65	83	15.502	5.735	0	0.527	0.281	48
88	83	15.523	5.743	0	0.663	0.413	49
146	83	15.491	5.742	0	0.457	0.392	53
195	83	15.476	5.742	0	0.364	0.392	55
293	83	15.520	5.742	0	0.647	0.389	57
439	83	15.490	5.743	0	0.452	0.409	62
645	83	15.515	5.767	0	0.612	0.830	65

Brightness: 255

cut off frequency (10^{-3} pixelunits^{-1})	N	mean (pixelunits)	σ (pixelunits)	ΔN	Δmean (%)	Δσ (%)	threshold value
8	83	15.465	5.731	0	0.291	0.204	94
12	83	15.452	5.725	0	0.204	0.107	96
14	83	15.458	5.729	0	0.242	0.162	99
18	83	15.440	5.728	0	0.125	0.148	102
29	83	15.435	5.726	0	0.093	0.118	103
48	83	15.447	5.729	0	0.170	0.170	103
65	83	15.462	5.736	0	0.269	0.291	105
88	83	15.451	5.729	0	0.200	0.163	107
146	83	15.453	5.727	0	0.213	0.140	110
195	83	15.446	5.729	0	0.168	0.161	112
293	83	15.448	5.731	0	0.182	0.198	115
439	83	15.460	5.729	0	0.257	0.170	119
645	83	15.464	5.737	0	0.284	0.301	123

MINMAX FILTER + ISODATA THRESHOLD

Brightness objects: trend
Size of filter: 31

Brightness: 127+0.125x

cut off frequency (10^{-3} pixelunits^{-1})	N	mean (pixelunits)	σ (pixelunits)	ΔN	Δmean (%)	Δσ (%)	threshold value
8	83	15.318	5.881	0	-0.665	2.831	51
12	83	15.358	5.884	0	-0.402	2.875	52
14	83	15.373	5.830	0	-0.276	1.941	56
18	83	15.407	5.782	0	-0.087	1.095	58
29	83	15.399	5.821	0	-0.140	1.777	59
48	83	15.390	5.817	0	-0.196	1.714	60
65	83	15.427	5.779	0	0.046	1.041	62
88	83	15.446	5.797	0	0.169	1.363	63
146	83	15.429	5.796	0	0.056	1.338	67
195	83	15.420	5.800	0	-0.005	1.410	69
293	83	15.460	5.789	0	0.256	1.220	71
439	83	15.414	5.795	0	-0.039	1.326	76
645	83	15.402	5.844	0	-0.116	2.177	79

Brightness: 127+0.250x

cut off frequency (10^{-3} pixelunits^{-1})	N	mean (pixelunits)	σ (pixelunits)	ΔN	Δmean (%)	Δσ (%)	threshold value
8	83	15.124	5.988	0	-1.920	4.691	66
12	83	15.189	5.937	0	-1.503	3.808	67
14	83	15.212	5.960	0	-1.354	4.202	71
18	83	15.230	5.945	0	-1.232	3.946	73
29	83	15.234	5.937	0	-1.211	3.805	74
48	83	15.229	5.917	0	-1.242	3.451	75
65	83	15.257	5.874	0	-1.060	2.707	77
88	83	15.281	5.887	0	-0.903	2.926	78
146	83	15.265	5.885	0	-1.004	2.888	82
195	83	15.277	5.905	0	-0.932	3.244	83
293	83	15.269	5.907	0	-0.981	3.289	86
439	83	15.272	5.898	0	-0.964	3.129	90
645	83	15.243	5.923	0	-1.151	3.566	94

UNIFORM FILTER + ISODATA THRESHOLD

Brightness objects: constant
Size of filter: 31

Brightness: 127

cut off frequency (10^{-3} pixelunits^{-1})	N	mean (pixelunits)	σ (pixelunits)	ΔN	Δmean (%)	Δσ (%)	threshold value
8	83	15.230	5.277	0	-1.233	-7.726	58
12	83	15.231	5.293	0	-1.231	-7.462	61
14	83	15.265	5.314	0	-1.009	-7.086	66
18	83	15.262	5.329	0	-1.024	-6.820	72
29	83	15.219	5.313	0	-1.306	-7.109	70
48	83	15.303	5.320	0	-0.758	-6.984	68
65	83	15.304	5.276	0	-0.755	-7.744	75
88	83	15.305	5.250	0	-0.745	-8.201	75
146	83	15.350	5.285	0	-0.456	-7.600	74
195	84	15.185	5.399	1	-1.523	-5.596	73
293	88	14.675	5.771	5	-4.835	0.902	77
439	102	13.199	6.522	19	-14.404	14.035	79
645	122	11.627	6.866	39	-24.601	20.041	83

Brightness: 255

cut off frequency (10^{-3} pixelunits^{-1})	N	mean (pixelunits)	σ (pixelunits)	ΔN	Δmean (%)	Δσ (%)	threshold value
8	83	15.246	5.322	0	-1.128	-6.946	125
12	83	15.242	5.319	0	-1.158	-6.992	127
14	83	15.255	5.329	0	-1.072	-6.820	127
18	83	15.261	5.333	0	-1.032	-6.755	129
29	83	15.269	5.339	0	-0.984	-6.647	127
48	83	15.268	5.341	0	-0.986	-6.609	126
65	83	15.277	5.322	0	-0.929	-6.950	126
88	83	15.299	5.321	0	-0.784	-6.962	126
146	83	15.309	5.328	0	-0.656	-6.849	127
195	83	15.268	5.323	0	-0.721	-6.921	129
293	83	15.264	5.329	0	-0.990	-6.829	131
439	83	15.249	5.332	0	-1.011	-6.765	133
645	83	15.243	5.331	0	-1.112	-6.790	135

UNIFORM FILTER + ISODATA THRESHOLD

Brightness objects: trend
Size of filter: 31

Brightness: 127+0.125x

cut off frequency (10^{-3} pixelunits^{-1})	N	mean (pixelunits)	σ (pixelunits)	ΔN	Δmean (%)	Δσ (%)	threshold value
8	83	15.169	5.294	0	-1.633	-7.440	84
12	83	15.186	5.293	0	-1.519	-7.460	85
14	83	15.208	5.300	0	-1.374	-7.337	89
18	83	15.223	5.302	0	-1.282	-7.294	91
29	83	15.182	5.292	0	-1.543	-7.474	94
48	83	15.188	5.301	0	-1.509	-7.309	94
65	83	15.223	5.321	0	-1.280	-6.957	96
88	83	15.211	5.307	0	-1.357	-7.217	96
146	83	15.242	5.315	0	-1.158	-7.061	99
195	83	15.234	5.305	0	-1.211	-7.239	96
293	83	15.230	5.317	0	-1.237	-7.032	98
439	83	15.234	5.318	0	-1.210	-7.015	101
645	83	15.194	5.302	0	-1.470	-7.304	108

Brightness: 127+0.250x

cut off frequency (10^{-3} pixelunits^{-1})	N	mean (pixelunits)	σ (pixelunits)	ΔN	Δmean (%)	Δσ (%)	threshold value
8	83	15.122	5.284	0	-1.935	-7.610	108
12	83	15.128	5.284	0	-1.895	-7.619	105
14	83	15.257	5.296	0	-1.707	-7.393	106
18	83	15.116	5.287	0	-1.975	-7.559	109
29	83	15.127	5.291	0	-1.900	-7.481	112
48	83	15.108	5.314	0	-2.023	-7.088	113
65	83	15.158	5.316	0	-1.703	-7.048	114
88	83	15.141	5.314	0	-1.809	-7.095	115
146	83	15.129	5.304	0	-1.888	-7.268	116
195	83	15.160	5.309	0	-1.687	-7.175	115
293	83	15.172	5.309	0	-1.612	-7.172	116
439	83	15.174	5.321	0	-1.595	-6.971	118
645	83	15.143	5.316	0	-1.798	-7.050	121

APPENDIX C

PROGRAMS

C. SHORT EXPLANATIONS OF THE COURSE OF THE PROGRAMS

C1. METHOD OF SECOND DERIVATIVE

PUBL2nd.tip

The operation of distinguishing objects by use of the second derivative is performed by use of the program *PUBL2nd.tip*, consisting of a series of TCLi-commands.

command:

lapl r1 r3

r1: original image (exposure)

r3: second derivative

For a given object, the second derivative as yielded in r3 looks like:

```
                 -5  -7  -3
     -3  -2  -2   4  -1  -6  -4  -3
 -6  -7  10   1  13   6   2   3  -2
 -3   2   4   7   5  11  -1  -8  -4
 -1  -2  -9  -5  -4   4   1  -2
                 -6  -1  -2
```

Segmentation is done with a threshold=1, by the TCLi command:

thresh r3 bt1 f 1

r3: second derivative;

bt1: bitplane;

f: fixed mode for the threshold value;

1 : threshold value.

For the object given aboven in bitplane bt2 this yields:

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

The pixel in bt2 gets value=1 if the 2^{nd} derivative>0, otherwise 0.

Reference areas:

The first derivative is determined with use of robert gradient filter, by the command:

robg r1 r4

r1: original image (exposure);

r4: first derivative.

Reference areas are obtained from the iterativ threshold. First the threshold is calculated by the self made command *thrmsgm*, based on nσ-method (see B 1.1):

thrmsgm r4 thr

r4: first derivative;

thr: threshold value.

And second the reference areas are got by use of:

thresh r4 bt3 f thr

r4: first derivative;

bt3:bitplane, containing reference areas;

f: fixed mode;

thr: threshold value.

Most of the reference areas have the form of a ring. The closed spaces within are filled up by self made TCLi command:

paint bt3

In later versions of TCLi program package or TIM there is a standard command for this.

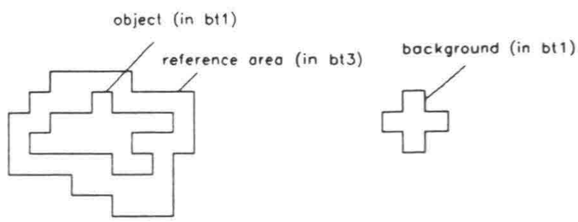
In some cases the threshold can be low enough for remainder background noise in bt3, usually consisting of 2 pixels or less. All objects of this size are removed by TCLi command:

bopen bt3 bt3 2

This TCLi command first erodes all objects in the image by two layers of pixels, and then dilates them again. Small objects that are disappeared after the erosion can not be dilated, so that they are removed from bt3. Except objects that are to be disappeared during running of this command the shapes and the surfaces do not change.

Selection of objects from the background.

The foreground pixels in bt1, representing locations of second derivative >0, do not only come from the objects but also from the background fluctuations (see below, and also chapter 3).



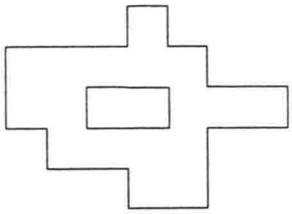
The object items fit in the reference areas, whereas the background items do not. Object items can be selected by use of binary and-operator:

band bt1 bt3 bt1

Then the bitplane bt1 consists of foreground pixels from objects only.

Finishing off

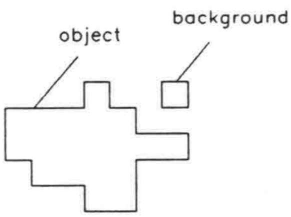
Not only the background fluctuates but it is also possible that the foreground fluctuates as a result of different shades on the particle surface. This causes holes in some objects in the bitplane bt1:



These holes are filled up with use of:

paint bt1

In some cases that areas of 2nd derivative>0 of the background noise lie so narrow to the objects so that a little part of about 1 or 2 pixels also fits in the reference area. After selection, they look like:



The number of the pixels of the background are very few and can be removed by use of TCLi command:

bopen bt1

The number of erosion cycles is not given after bopen. This means that it occurs in one erosion cycle (default).

In the moist exposures one or more objects are connected to the edge of the image. These give a wrong image of the surfaces and therefore an extra deviation in the determination. These objects are removed by use of:

remedcon bt1

Remedcon.tip is a self made TCLi-batch program which removes objects that are connected to the edge of the image. Like paint.tip in later version of TCLi program package or TIM there is a standard command for this.

C 1.1 Iterative threshold program *thrnsgm*

The threshold for segmentation of the reference areas is determined iterativ by use of *thrnsgm* a TCLi-command of a self made C-program based on nσ-method: threshold = mean + nσ.
The σ is the RMS value of the background noise and is determined with:

$$\sigma = \sqrt{\overline{x^2} - \overline{x}^2}$$

in which

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\overline{x^2} = \frac{1}{N}\sum_{i=1}^{N} x_i^2$$

x_i: value of background pixel
N: total number of background pixels

The first derivative image consists of pixels of background noise and pixels of signals caused by the boundaries between objects and background. For determination of the threshold the pixels of the background are selected and this occurs iterativ.

<u>Iteration process</u>

In the first cycle, the mean and the sigma are calculated over all pixels of the first derivative image. By calculating also the pixels of object boundaries, the mean and the sigma are greater than those of the noise only. In the second cycle, they are calculated again, but now the pixels greater than mean + nσ are not used. This is repeated until the difference of the last mean + nσ with the previous one is two or less. (After this, the iteration process advances too slowly.) Now the mean and the sigma are calculated over the pixels of background noise only.

For the iteration process the value of n is chosen 3. For background fluctuations, which have a probability density function in a shape of normal distribution, 0.13% of the pixels of the background fluctuations exceeds mean + nσ. However, the real shape of the function is inpredictable. Therefore, the value of n that is used for segmentation is chosen twice as large: 6 instead of 3.

The program

Calculating mean and σ is simplified by using a histogram of the first derivative image. Therefore *thrnsgm* starts with determination of the histogram, from which is calculated (1) the mean over all pixel values in the first derivative image:

$$\bar{x}=\frac{1}{N}\sum_{i=1}^{N}x_i=\frac{1}{N}\sum_{i=0}^{max}h_i i$$

N: total number of pixels: 512^2=262144
x_i: pixel value
h_i: number of pixels with value i (histogram)
max: maximum pixel value.

And (2) the mean over all squared pixel values:

$$\overline{x^2}=\frac{1}{N}\sum_{i=1}^{N}x_i^2=\frac{1}{N}\sum_{i=0}^{max}h_i i^2$$

For calculation the next mean + nσ, the pixels greater than this mean + nσ are not used:

$$\bar{x}=\frac{1}{N_j}\sum_{i=1}^{N_j}x_i=\frac{1}{N_j}\sum_{i=0}^{thr}h_i i$$

$$\overline{x^2}=\frac{1}{N_j}\sum_{i=1}^{N_j}x_i^2=\frac{1}{N_j}\sum_{i=0}^{thr}h_i i^2$$

thr: integer (≤ mean + nσ)
j = 1,2,...,k, is the number of the cycle

and:

$$N_j=\sum_{i=1}^{thr}h_i$$

C2. METHOD OF UNIFORM FILTERING + ISODATA THRESHOLD

PUBLuni.tip

The objects distinguished by use of uniform filtering, followed by isodata threshold is performed by use of the program *PUBLuni.tip*, consisting of a series of standard TCLi-commands.

The original image filtered by a uniform filter by use of the command:

unif r1 r2 size
r1: original image (exposure)
r2: image filtered uniformly
size : size of the moving window

The image is corrected on shading by use of:

sub r2 r1 r3
r1: original image (exposure)
r2: image filtered uniformly
r3: corrected image

Because of the use of the histogram, isodata threshold operates in the range of 0..255 grey values, therefore the values over all pixels in the corrected image have to be in this range.
It is possible that the corrected image r3 has also pixels with a negative value. To avoid this, two commands are used:

minval r3 mn
sub mn r3
r3: corrected image
mn: minimum value over all pixels

The minimum value over all pixels will be 0. The maximum value however can exceeds 255 grey values. Be back to 255 if the maximum is greater than 255 contrast stretching command *cst* is used:

if (mx >= 255) then
cst r3
endif

The minimum value remains 0.

Then the objects are distinguished with use of isodata threshold command:

thresh r3 bt1 i
r3: corrected image
bt1: bitplane consisting of distinguished objects.
i: isodata mode

And finally objects that are connected to the edge of the image, are removed by use of the command:

remedcon bt1
bt1: bitplane consisting distinguished objects.

C3. METHOD OF MINMAX FILTERING + ISODATA THRESHOLD

PUBLmimxi.tip

The objects distinguished by use of minmax filtering, followed by isodata threshold is performed by use of the program *PUBLmimxi.tip*, consisting of a series of standard TCLi-commands.

The minmax filter, better say local minmax filter, is split up into local minimum filter and local maximum filter, so that the original image is filtered by a minimum filter and the output in turn by a maximum filter:

- local minimum filtering:

  lmin r1 r2 size
  r1: original image (exposure)
  r2: local minimum filtered image
  size : size of the moving window

- local maximum filtering:

  lmax r2 r3 size
  r2: local minimum filtered image
  r3: local minmax filtered image
  size : size of the moving window

The image is corrected on shading by use of substraction:

sub r3 r1 r4
r1: original image (exposure)
r3: local minmax filtered image
r4: corrected image

It is impossible that the minimum over all pixels of corrected image is below zero and the maximum exceeds 255, so that isodata threshold can be used directly:

thresh r4 bt1 i
r4: corrected image
bt1: bitplane consisting distinguished objects.
i: isodata mode

And finally for objects that are connected to the edge of the image, they are removed by use of the command:

remedcon bt1
bt1: bitplane consisting distinguished objects.

APPENDIX D

PROGRAM LISTINGS

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!                                                                         !!
!!                minmax filtering + isodata threshold                     !!
!!               --------------------------------------                    !!
!!                                                                         !!
!!      Program: PUBLmimxi.tip                                             !!
!!                                                                         !!
!!      Distinguish objects from background by use of                      !!
!!      minmax filter and isodata threshold.                               !!
!!                                                                         !!
!!      Objects which are at the edge of the image                         !!
!!      are removed.                                                       !!
!!                                                                         !!
!!      The results are stored in a bitplane.                              !!
!!                                                                         !!
!!      Input parameters                                                   !!
!!        argument 1: exposure or original image (512*512 short integer)   !!
!!        objects: light, background dark.                                 !!
!!        argument 3: size of the filter                                   !!
!!      Output parameter or argument 2:                                    !!
!!        distinguised objects (512*512 sbit, first bitplane in r1).       !!
!!                                                                         !!
!!      r1 consists of 8 bitplanes: bt1, bt2, ..., bt8.                    !!
!!                                                                         !!
!!      ! The pixel values of r1 are changing during running of            !!
!!        this program.                                                    !!
!!                                                                         !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
declare/arg=1 r1                        ! original image
declare/arg=2 bt1                       ! destiguised objects
declare/arg=3 size                      ! size of the filter
declare r2 short 512 512                ! minimum filtered image
declare r3 short 512 512                ! shading image
declare r4 short 512 512                ! corrected image

lmin r1 r2 size         ! local minimum filtering
lmax r2 r3 size         ! local maximum filtering
sub r3 r1 r4            ! correction
thresh r4 bt1 i         ! isodata threshold
remedcon bt1            ! remove edge connected objects
kill *
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!                                                                        !!
!!                 uniform filtering + isodata threshold                  !!
!!                 -------------------------------------                  !!
!!                                                                        !!
!!      Program: PUBLuni.tip                                              !!
!!                                                                        !!
!!      Distinguish objects from background by use of uniform             !!
!!      filter and isodata threshold.                                     !!
!!                                                                        !!
!!      Objects which are at the edge of the image                        !!
!!      are removed.                                                      !!
!!                                                                        !!
!!      The results are stored in a bitplane.                             !!
!!                                                                        !!
!!      Input parameters:                                                 !!
!!        argument 1:exposure or original image (512*512 short integer)   !!
!!        objects: light, background dark.                                !!
!!        argument 3: size of the filter                                  !!
!!      Output parameter or argument 2 :                                  !!
!!        distinguished objects (512*512 sbit, first bitplane in r1).     !!
!!                                                                        !!
        r1 consists of 8 bitplanes: bt1, bt2, ..., bt8.                   !!
!!                                                                        !!
!!      ! The pixel values of r1 are changing during running              !!
!!        of this program.                                                !!
!!                                                                        !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

declare/arg=1 r1               ! original image
declare/arg=2 bt1              ! destinguised objects
declare/arg=3 size             ! size of the filter
declare r2 short 512 512       ! shading image
declare r3 short 512 512       ! corrected image
decl mn short; decl mx short

unif r1 r2 size        ! uniform filtering
sub r2 r1 r3           ! correction
minval r3 mn           !     ,,
sub mn r3              !     ,,
maxval r3 mx           !     ,,
if (mx >= 255) then    !     ,,
 cst r3                !     ,,
endif                  !     ,,
thresh r3 bt1 i;       ! isodata threshold
remedcon bt1           ! remove edge connected objects
kill *
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!                                                                           !!
!!                           second derivative                               !!
!!                          -------------------                              !!
!!                                                                           !!
!!       Program: PUBL2nd.tip                                                !!
!!                                                                           !!
!!       Distinguish aeras of the second derivative >0, by the objects.      !!
!!                                                                           !!
!!       Objects which are at the edge of the image are removed.            !!
!!                                                                           !!
!!       The results are stored in a bitplane.                               !!
!!                                                                           !!
!!       Input parameter:                                                    !!
!!         argument 1: exposure or original image (512*512 short integer)   !!
!!         objects: light, background dark.                                  !!
!!       Output parameter:                                                   !!
!!         argument 2: binaire image consisting distinguised objects.       !!
!!                      (512*512 sbit, first bitplane in r1)                 !!
!!                                                                           !!
!!       r1 consist of 8 bitplanes: bt1,bt2,...,bt8.                         !!
!!                                                                           !!
!!       ! The pixel values of r1 are changing during running of            !!
!!         this program.                                                     !!
!!                                                                           !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

decl/arg=1 r1                ! original image
decl/arg=2 bt1               ! destinguised objects
declare r3 short 512 512     ! second derivative image
declare r4 short 512 512     ! first derivative image
declare thr int              ! threshold value

lapl r1 r3                   ! determining 2nd deriv. with laplace filter
thresh r3 bt1 f 1            ! determining areas of 2nd deriv. > 0
robg r1 r4                   ! determining first derivative
thrnsgm r4 thr               ! calculating threshold (iterativ)
thresh r4 bt3 f thr          ! determining reference areas
paint bt3                    ! fill up closed spaces
bopen bt3 bt3 2              ! remove remainder background noise
band bt1 bt3 bt1             ! select objects from background
paint bt1                    ! finish off
bopen bt1                    !    ,,
remedcon bt1                 !    ,,      ,remove edge connected objects
kill *                       !    ,,
```