

Use of a computer for evaluation of flow films

Citation for published version (APA):

Lemmens, W. J. M. (1985). *Use of a computer for evaluation of flow films*. (Computing science notes; Vol. 8503). Technische Hogeschool Eindhoven.

Document status and date:

Published: 01/01/1985

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Use of a Computer for
Evaluation of Flow Films

by

W.J.M. Lemmens

85/03

COMPUTING SCIENCE NOTES

This is a series of notes of the Computing Science Section of the Department of Mathematics and Computing Science of Eindhoven University of Technology.

Since many of these notes are preliminary versions or may be published elsewhere, they have a limited distribution only and are not for review.

Copies of these notes are available from the author or the editor.

Eindhoven University of Technology
Department of Mathematics and Computing Science
P.O. Box 513
5600 MB EINDHOVEN
The Netherlands
All rights reserved
editor: M.L. Potters

Use of a computer for evaluation of flow films.

W.J.M. Lemmens.

Abstract.

One of the techniques used for studying turbulent flows in water uses electrolytically generated markers. These are recorded on film as they move with the flow. Up till now, these films had to be evaluated by hand.

We now have developed a technique that can be used to do this evaluation by computer, using a TV-camera and electronics for digitising the image. This technique is based on the projection of part of the image on the X and Y - axes and using the arrays of projected values to find the X and Y - coordinates of the bubble blocks.

In the paper, the technique used will be elaborated and some results will be presented. Also, some more uses of this efficient data reduction technique will be indicated.

I. Introduction.

There are many experiments in physics that yield images as a result. Notable examples are bubble chamber photographs in high energy physics and, as in our case, cine films or video recordings made to study the movements of all kinds of objects. Often, the pictures made have to be evaluated further to obtain the relevant data of the experiment. Especially if there are many pictures to be evaluated because of low experimental yields or with long stretches of cine film, the amount of work associated with this kind of evaluation becomes prohibitive. That is why more and more computer assistance is sought for this job.

Our problem was to devise a system for the extraction of meaningful data from a cine film of markers

moving along with a (turbulent) flow of water. The computer would have to process frame after frame of the film and deliver the coordinates of those markers that appear within the area of interest. The markers appear as light stains against a darker background. At the point where they originate they are approximately rectangular in shape, but as they move they are progressively deformed (fig. 1). They are generated by electrolytical formation of blocks of hydrogen bubbles at a thin platinum wire that has alternatively isolated and unisolated parts, through the application of rectangular voltage pulses (1).

The markers in a picture are laid out in a roughly rectangular pattern, with rows of markers parallel to the X - axis of the picture and columns parallel to the Y - axis. The adherence to this pattern is not perfect. In fact, it is the deviation from the regular pattern we are interested in. However, the semi-regular structure of the picture offers a usefull search strategy, that enables us to speed up the evaluation process considerably.

Direct evaluation of images by computer is hampered by the amount of data that one picture represents. A typical picture as used for digital processing contains some 65536 image points (or picture elements, pixels), each of which is represented by an 8-bit binary number. In total, that is 524288 bits of data. As the amount of information to be derived from the picture can mostly be measured in hundreds of bits, this represents quite a large redundancy factor. There are many techniques for data reduction in general use, most of which start by thresholding the image, thereby reducing the number of gray values to two (black and white), or the number of bits per pixel to only 1, followed by the determination of characteristic quantities or the selection of points of interest (2). These are general purpose methods that can be used on any picture. After removing many superfluous bits in this way, the picture will have to be

analysed, using a description of its contents or its structure, in order to get the data one needs. One could call this the model-driven part of image processing.

The method we present here actually works the other way around, the model of the picture driving the data reduction process. In order to obtain the 60 or so coordinate pairs we needed from the markers within the area of interest, we used a stepwise approach, in which a general purpose data reduction technique was applied successively to areas of the picture that were determined by a model of the composition of the picture. Also, our method does not depend on thresholding, but uses all gray values that are present.

II. The search process.

In order to find the position of the individual markers in a flow picture efficiently, we should use as much knowledge of the pictures as is available. This knowledge should result in a choice of method and a way of using it that are optimal for the problem at hand. An examination of the pictures to be analysed reveals the following:

1. The position of a bubble block may in first approximation be described as the intersection of a horizontal and a vertical line in a rectangular grid, with some added deviation:

$$x_{ik} = x_0 + i \cdot d_x + (\Delta x)_{ik}$$

$y_{ik} = y_0 + k \cdot d_y + (\Delta y)_{ik}$, with x_0 , y_0 , d_x , and d_y being constant, and Δx and Δy having a mean value of 0. An example derived from an actual flow picture is given in table 1.

2. Inside the area of interest, blocks do not overlap. Between any two white blocks, there is a dark area.

3. Blocks are roughly rectangular in shape, with the lines separating them being narrower than the blocks themselves.

Inspired by these observations, we devised a strategy of finding each block in a number of steps:

1. Determine the area of interest and find reference points for calculating the coordinate transformation matrix.

A frame from the flow film contains areas, such as the one before the point at which the markers are generated, that supply hardly any information. Further, there are areas, such as the immediate vicinity of the vessel wall and the parts with heavy turbulent disturbance of the flow, in which the image of the markers is so garbled as to make it virtually impossible to gain useful information there. These parts should be excluded from further analysis.

There are also points in the image that do not contribute to the information on the flow of liquid, but that are important for deriving information on the geometry of the recorded pictures. Examples are the position of the wire at which the markers originate and the position of the vessel wall. Knowledge of the position of these points, both in picture coordinates and in the real world, allows the picture coordinates of the markers to be converted into world coordinates.

2. Determine the position of successive rows of markers.

The geometry of the pictures is such that the markers form a file of rows, each row originating at a specific point along the wire that is suspended in the flow. These rows undulate and twist in a highly unpredictable manner, but within certain bounds. By using the method described in the next section, we can find some average position of a row of markers in the vertical direction, with

some indication of upper and lower limits.

3. Find some approximation of the positions of markers within a row.

Every row of markers is composed of individual, quasi-rectangular spots that appear at a varying distance from each other. Again, this variation occurs within certain bounds. Apart from the rows of markers, in a flow picture one can also discern (distorted) columns of markers. The same method as used in 2. is applied to subdivide a row of markers into individual marker areas.

4. Within a limited area surrounding the positions found in 2. and 3., determine the exact position of a marker.

Finding the boundaries of a specific object in an average picture poses special problems. The overall gray value of the objects to be found may vary with their position and even from one point to another within the object, and the transitions from object to background may not be equally sharp everywhere. For that reason we made no attempt to find the boundaries of the markers. Instead, the position of a marker is defined as the centre of gravity of the surrounding rectangle, whose sides are parallel to the axes. Therefore some measure of the minimum and maximum X and Y - values of a marker has to be found that is sufficiently precise and that may be determined sufficiently efficiently, in order to establish this rectangle.

Again, the standard data reduction method described hereafter provides the answer.

III. The data reduction method.

In the introduction we mentioned thresholding as a way to reduce the amount of data of raster scan images. There are a number of other methods, many of which use far

more complicated algorithms, that take a lot of computer time. The method we chose, however, is simple and fast like thresholding. It works by projecting part of the image on the axes, yielding two arrays, P_x and P_y , of projected values:

$$P_{x_i} = \sum_{j=y_0}^{y_m} f_{ij}, \quad x_0 \leq i \leq x_m,$$

$$P_{y_j} = \sum_{i=x_0}^{x_m} f_{ij}, \quad y_0 \leq j \leq y_m,$$

with f_{ij} = gray value of pixel (i,j) .

Figs. 2 to 5 show some windows used in the bubble pictures and some examples of the projections obtained from them.

This method has a number of usefull features:

1. It provides data reduction. An image of $N \times M$ gray values is reduced to $N + M$ values. For a full picture this amounts to a reduction from 524288 to 8192 bits.
2. It suppresses noise. The variance of the noise of the mean of N points is $1/N$ x the variance of the noise of a single point (3).
3. It reduces the problem dimensionality and thereby the complexity of the solution. A two-dimensional problem is reduced to two one-dimensional problems. Finding a specific object can now be done for the X- and Y-directions separately.

There is, however, the possibility that the features to be found will in the X- or Y-projection be masked by an object with the same position in the Y- or X-direction. Interference by other objects may be prevented by narrowing

the projection window in order to exclude those objects from the projection.

4. It can easily be realised in hardware and thereby be made even cheaper and faster.

The use of windows considerably facilitates a top-down analysis of an image: A projection of the whole image provides indications of coarse image features. It shows one where to look and where not to look for further information. If the image has a known structure, as in our case, we can identify global features of that structure and set one or more windows that cover only the area of interest. Within those new windows new projections may be made to gain information on finer structural details, and so on. As windows get smaller, more disturbing objects will be excluded. This is clearly illustrated in fig. 3, in which most of the disturbance that occurs in fig. 2 has been suppressed by leaving out the parts that interfere. The projections will also reveal more detail, but noise effects will increase too.

Ultimately, in the case of a 2×2 array, the projection is an exact representation of the actual image points. That is, the original values may exactly be calculated from the projections and there is no question of data reduction at all. With larger windows one can obtain only an estimate of the original picture from the projections, by using the same techniques as in computer assisted (X-ray) tomography (4).

IV. Finding the positions of the bubble blocks.

Having reduced the problem of finding the individual blocks of bubbles to a one dimensional search in two directions we should find criteria to draw the line between

one row and the next and between the successive blocks within one row. One way to do so would be to find the local minima within a projection array. Any dark area between rows or columns of blocks would show as a minimum in the projection. Indeed, for larger projection windows, as used while searching for the different rows, this method proved to yield reliable results. But as projection windows decrease in size during the search for the individual blocks, noise is playing an increasingly important role. It causes false minima to appear, especially at the flat tops of the projected bubble blocks. So, in determining the final position of a block from a projection covering only a slightly larger area than the block itself, a different method has been used.

As only one full block is supposed to reside within the projection window, for each direction only two boundaries should be detected. The most reliable indication of the boundaries in this case proved to be the minima that were bounded by the monotonously non-decreasing or non-increasing sequence of points that had the largest difference between lowest and highest value. The first boundary should be the one with the largest difference between the minimum value at the boundary itself and the (local) maximum directly following it, provided it occurs before the second boundary. The second boundary should be the one with the largest difference between its projection value and the preceding maximum, provided it occurs after the first boundary. As the height of the mountain in-between is taken the minimum of the two slope heights. The relevant procedure is shown in fig. 6. As fig. 5 shows, taking the two lowest minima not always gave the correct bubble position.

As the final position of a block is taken the point exactly halfway between the two boundaries defined above. This is not always the maximum of the curve between those

points. The choice of the maximum for the final position would not yield reliable results, as due to its flat-topped shape the maximum of the curve would be rather sensitive to noise effects.

V. Choice of parameters.

After we have established a method for finding bubble blocks there are still a number of problems to be solved and questions to be answered before it will be fully operational. First, the pictures we have to deal with do contain stray bubbles. These are mostly due to the flow disturbances caused by the platinum wire that generates the bubble blocks. The stray bubbles appear as white dots in the picture, in between the blocks we are looking for, where they give rise to projection values comparable to those of the blocks themselves. A histogram of the dimensions of objects found in a picture shows them as a small peak to the left of the peak from the real blocks (fig. 7). So they may easily be removed from the results by rejecting all objects with dimensions smaller than a certain threshold.

Our second problem is the spread of the dimensions of the blocks themselves. As the histograms show, the bubble blocks may differ in their X- or Y-dimension by a factor of two. So using fixed-size windows for the final determination of the position of the blocks is out of the question. Window size should be variable and a little larger than the dimensions derived from the preceding X- and Y-projections. How much larger does depend on the third complicating factor, the deviation of the position of individual blocks from the average established using the Y-projection of the area of interest. Fig. 8 shows the deviation of the final position established in step 4 from the preliminary Y- and X-values from steps 2 and 3 in section II.

So we have two parameters that should be adjusted

for optimal results: The minimum size of an object to be recognized as a genuine bubble block, and the enlargement factor of the window used for precisely determining the coordinates of a block. As we have shown, these parameters may easily be derived from some measurements.

Our main concern was not to conduct all kinds of flow experiments, but to devise a method to evaluate films from routine experiments. So we did no in-depth investigation of all possible kinds of flow film. However, with the sample pictures we used to test our method, we found no false hits and no false misses, once the parameters had been properly adjusted.

VI. Other uses of the projection technique.

The method of using X- and Y-projections of partial pictures seems the perfect method for evaluating pictures of bubble blocks in a flow. The rectangular form of the windows used for projection neatly coincides with the form of the markers and the wanted coordinates may easily be derived from the projections. However, the method has also been applied to other image processing tasks. In a number of experiments we have used it for a wide range of purposes, such as comparing pictures in general and detecting and locating specific objects in an image.

Two images may be compared by subtracting one from the other, possibly after data reduction, and grading the difference according to some well-chosen criterion. We used this to detect and locate changes in a scene in front of a camera by making projections at regular intervals and calculating the modi of the differences between the new X-projection and the previous one, and the new Y-projection and the one preceding it. The difference-of-projection arrays so obtained were used in much the same way as the

projection arrays in the method described above, to find new objects appearing in the scene.

In order to find a known object in an image by using projections, first the signature (5) of that object should be determined. This is done by fitting a window closely around the image of the object and making the projections of the points within it. Now the characteristic pattern so obtained may be used to find the position of the same object in the projections made from a different picture. This may be done by one-dimensional template matching techniques:

$$Mx_i = \sum_k O_{x_k} \cdot Px_{i+k} \quad ,$$

$$My_j = \sum_l O_{y_l} \cdot Py_{j+l} \quad .$$

The positions of the object to be found is the position (i, j) for which Mx and My are a maximum.

This method has been used to track objects moving in front of all kinds of background. Of course, the method is very sensitive to the rotation and magnification of the object. So the best results are obtained if the stored signature of the object is updated every time the object is located, thus compensating for changes in view direction, distance and orientation. The time it takes to find an object in this way is linearly dependent on the dimensions of the search window, provided the projections themselves are made by hardware in real-time.

In our research we also tried other search strategies and other kinds of signature for identification of objects, such as the differences between successive projection values:

$$Px_i = \sum_j f_{ij} - \sum_j f_{(i-1)j}$$

$$P_{y_j} = \sum_i f_{ij} - \sum_i f_{i(j-1)}$$

Use of these values instead of the projections proper resulted in a more reliable recognition of the search objects.

These experiments have yielded encouraging results and certainly invite further investigation of the use of this technique as an image processing tool especially for time-critical applications, like robotics.

VII. Conclusions.

The conclusions we draw from the foregoing considerations may be summarized as follows:

1. Projections of partial images on the X- and Y-axes represent an efficient technique for routine use in computer evaluation of films of a liquid flow with bubble block markers.
2. The technique can be made to work in other applications too. It holds promises as a method to be used in robot-vision systems, especially if special hardware is built for it.
3. It deserves further research in order to establish the full extent of the application areas and the inherent limitations.

References:

- (1) F.A. Schraub et al., "Use of hydrogen bubbles for quantitative determination of time-dependent velocity fields in low-speed water flows." *Jrnl. Basic Engineering* 87 (1965), pp 429-444.
- (2) Gerald J. Agin, "Computer vision systems for industrial inspection and assembly." *IEEE Computer*, 13 (1980) 5, pp 11-20.
- (3) Z.Q. Wu, A. Rosenfeld, "Filtered projections as an aid in corner detection." *Pattern Recognition* 16 (1983), pp 31-38.
- (4) Ernest L. Hall, "Computer image processing and recognition." New York, Academic Press, 1979, pp. 258-304.
- (5) D.H. Ballard, C.M. Brown, "Computer vision." Englewood Cliffs, N.J., Prentice-Hall, 1982.

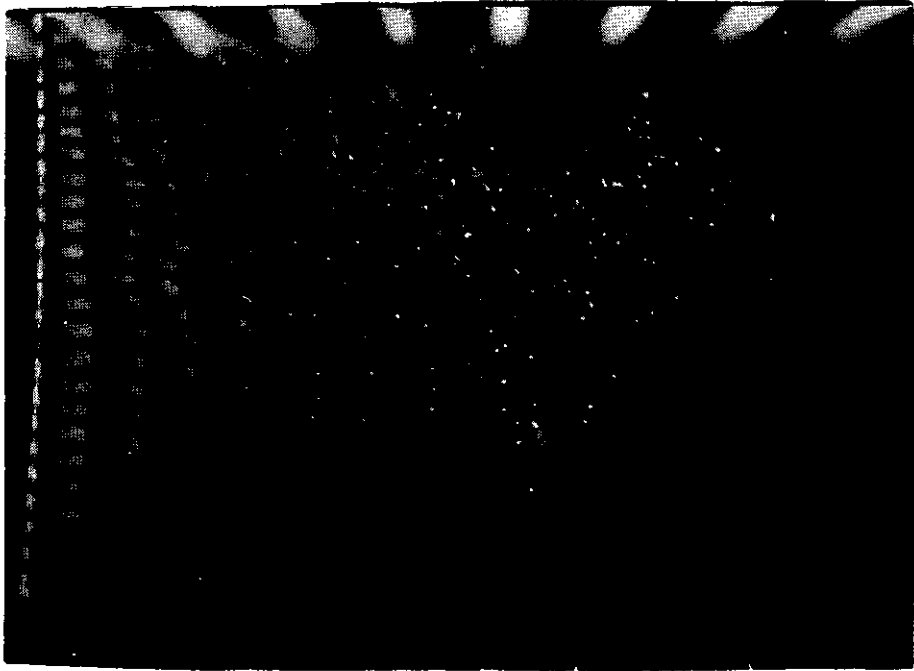


Fig. 1. Example of flow picture.

Mean X-values.

	54	66	80	94
37	(-3, -1)	(-5, -1)	(-9, -1)	(-11, 2)
46	(-2, -1)	(-5, -1)	(-9, 0)	(-13, 2)
54	(-2, -1)	(-5, 0)	(-9, 1)	(-12, 1)
62	(-2, -1)	(-4, -2)	(-6, 1)	(-10, 1)
70	(0, -1)	(-2, -2)	(-6, 0)	(-8, 2)
78	(0, -1)	(-2, -2)	(-6, 1)	(-6, 2)
87	(0, -2)	(-1, -3)	(-4, 1)	(-6, 4)
95	(1, -1)	(0, -2)	(-2, 0)	(-1, 2)
103	(1, -1)	(1, 0)	(0, 0)	(-1, 1)
111	(1, 0)	(1, 0)	(2, 0)	(0, 0)
119	(1, 0)	(2, 0)	(2, 0)	(3, 0)
128	(1, -1)	(2, 0)	(3, 0)	(5, 1)
138	(1, -1)	(1, 0)	(3, 1)	(4, 0)
148	(1, -2)	(2, -1)	(3, 1)	(5, 1)
157	(1, -2)	(3, -2)	(4, 1)	(9, 3)
165	(2, -2)	(3, -2)	(4, 2)	(9, 2)
173	(1, -1)	(3, -1)	(6, 1)	(10, 0)
182	(1, -1)	(4, -1)	(6, 1)	(10, 1)
191	(0, 0)	(3, -1)	(4, 0)	(8, 1)
201	(1, 0)	(3, -1)	(4, 0)	(8, 2)

Table: Mean X-values per column and Y-values per line and the deviations (ΔX , ΔY) from the mean X- and Y-values of the separate blocks in a typical picture.

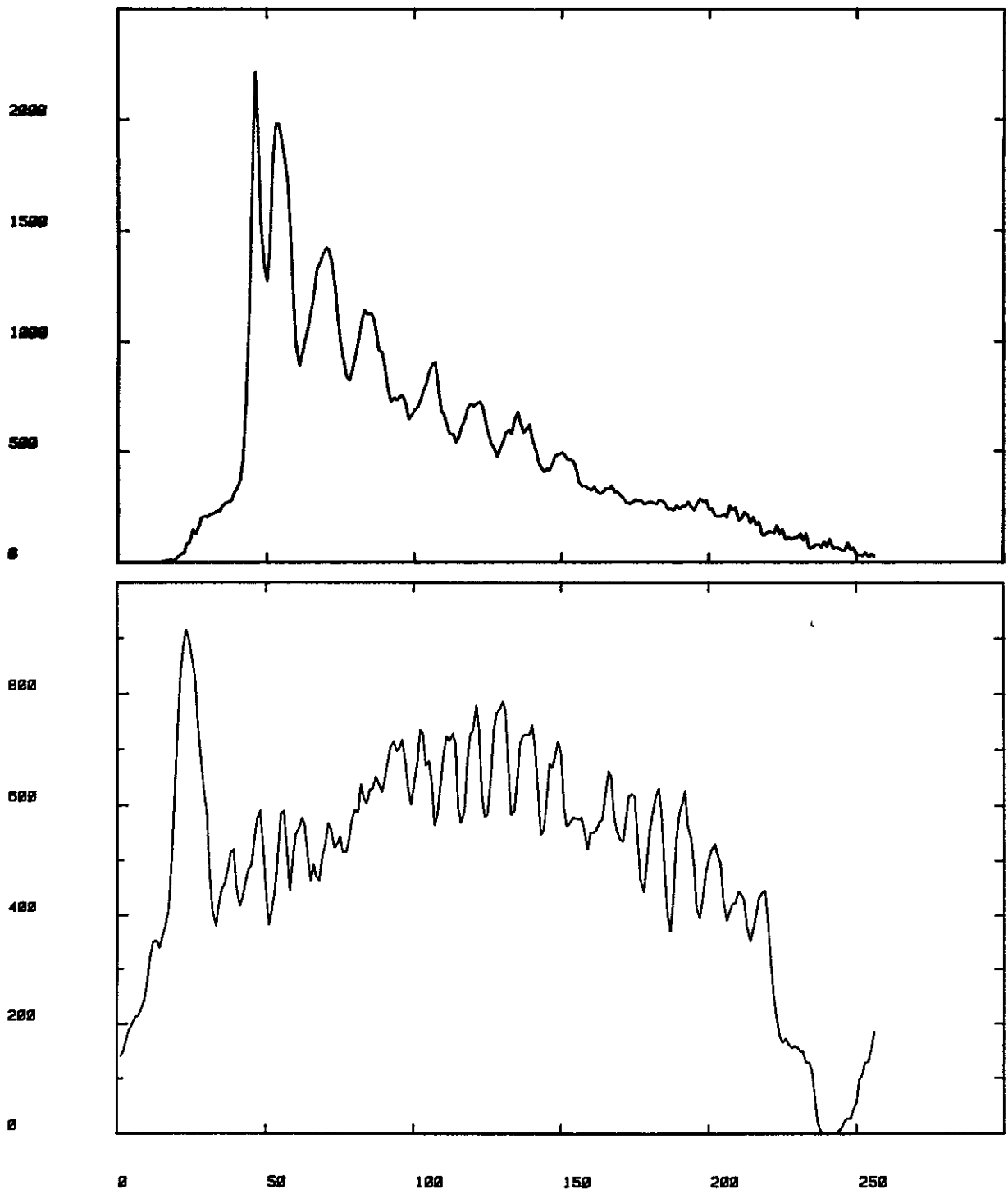


Fig. 2. X- and Y-projections of full picture.

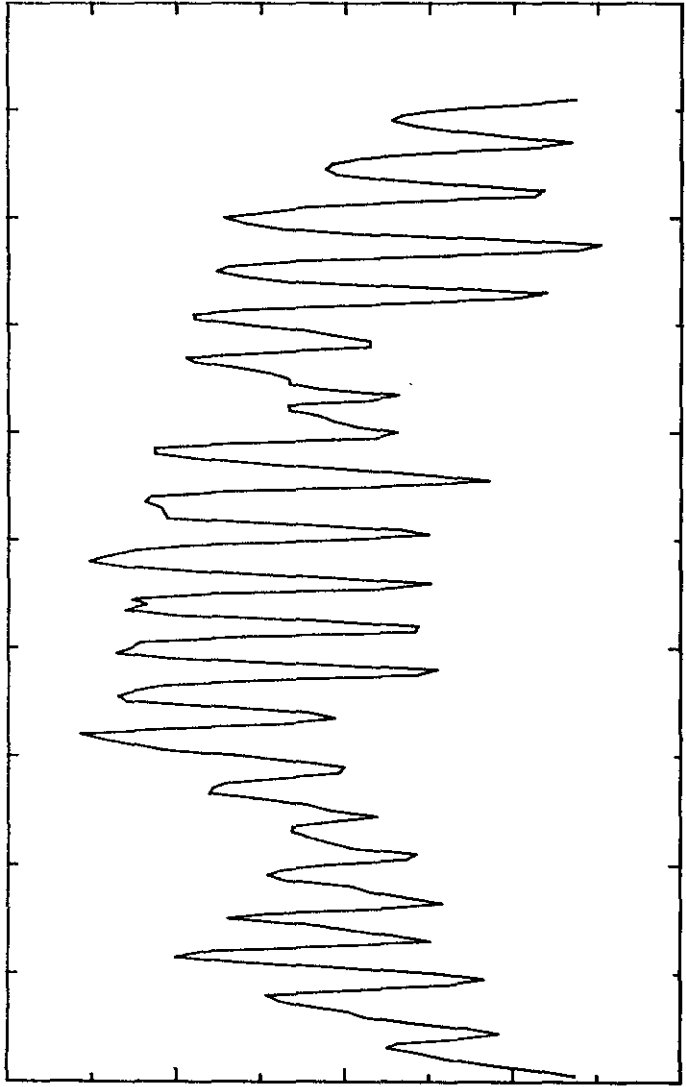
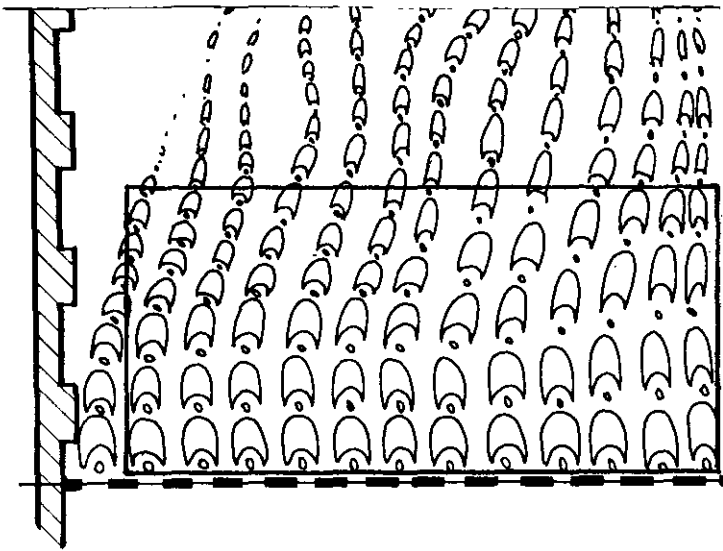


Fig. 3. Y-projection of area of interest.

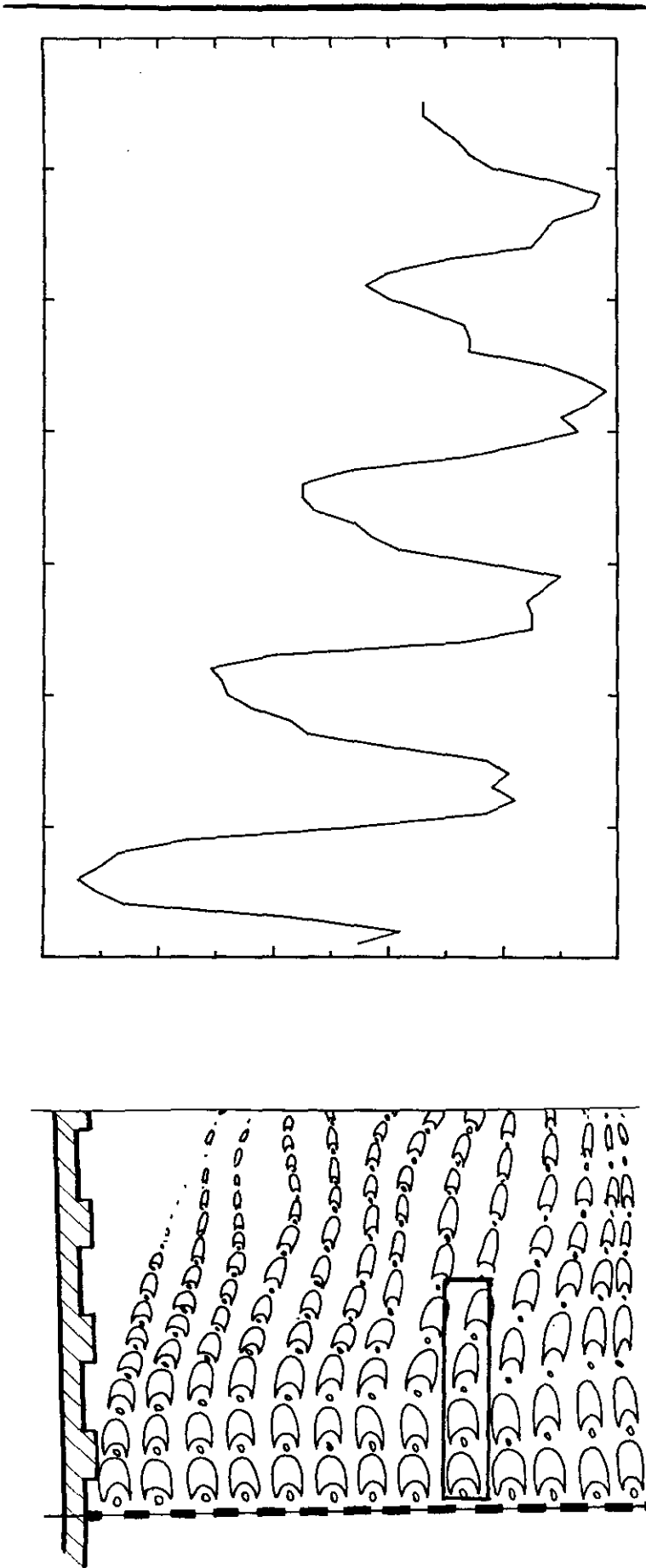


Fig. 4. X-projection of line area.

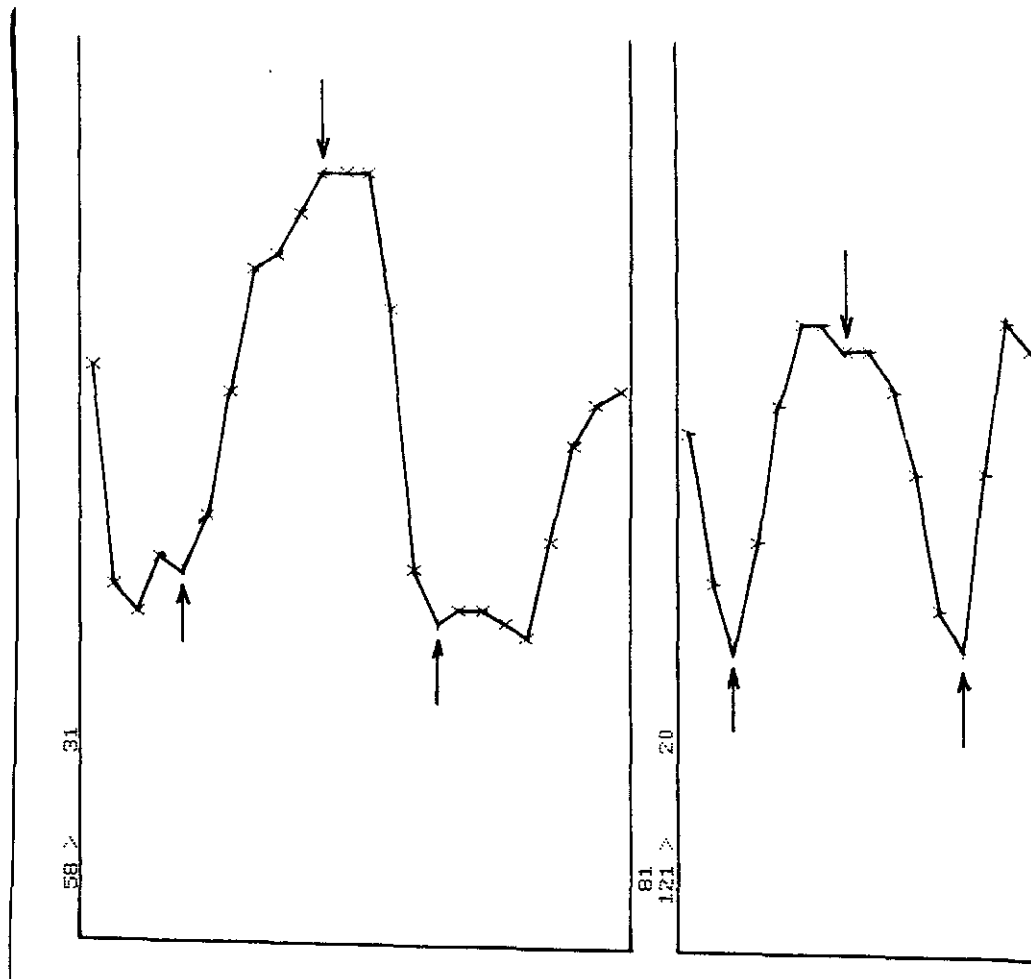
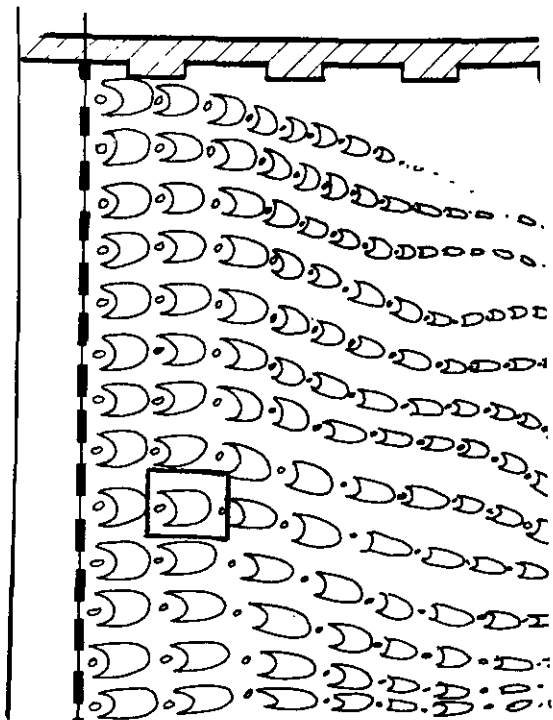


Fig. 5. Projections of single bubble block and immediate environment. The lower arrows indicate the limits of the block as found by the program. The upper arrow indicates the final position. Left: X-projection, right: Y-projection.

```

1
2 CONST MAXPXLIN=255;
3 TYPE LINRNG=0..MAXPXLIN;
4   PROJECTION=ARRAY[LINRNG] OF INTEGER;
5
6 PROCEDURE FINDTOP(XPROJ:PROJECTION; VAR XO,XM:LINRNG);
7 VAR I,MAXP,MINP,SOUSL,SOM,EOM:LINRNG; XP,XP1,MAXV,MINV,USLOPE,HOM:INTEGER;
8
9 BEGIN XP1:=XPROJ[XO]; MAXV:=XP1; MAXP:=XO; MINV:=XP1; MINP:=XO;
10  USLOPE:=0; SOUSL:=0; HOM:=0; SOM:=XO; EOM:=XM;
11  FOR I:=XO+1 TO XM DO
12  BEGIN XP:=XPROJ[I];
13    IF XP<XP1
14    THEN BEGIN IF MAXV=XP1
15      THEN (* LOCAL MAXIMUM *)
16        IF MAXV-MINV>USLOPE
17        THEN BEGIN USLOPE:=MAXV-MINV; SOUSL:=MINP
18          (* USLOPE = MAXIMUM UPWARD SLOPE SEEN SO FAR *)
19        END;
20        MINV:=XP; MINP:=I
21    END
22    ELSE IF XP>XP1
23    THEN BEGIN IF MINV=XP1
24      THEN (* LOCAL MINIMUM *)
25        IF MAXV-MINV>=USLOPE
26        THEN (* NEW DOWNWARD SLOPE >= HIGHEST UPWARD SLOPE SO FAR *)
27        BEGIN IF USLOPE>HOM
28          THEN BEGIN HOM:=USLOPE; SOM:=SOUSL; EOM:=I-1 END
29        END
30        ELSE IF MAXV-MINV>HOM
31        THEN (* UPWARD SLOPE > DOWNWARD SLOPE > CURRENT HEIGHT *)
32        BEGIN HOM:=MAXV-MINV; SOM:=SOUSL; EOM:=I-1 END;
33        (* CURRENT HEIGHT = MIN ( UPWARD SLOPE, DOWNWARD SLOPE ) *)
34        MAXV:=XP; MAXP:=I
35    END
36    ELSE IF MINV=XP
37    THEN (* MONOTONOUSLY NON-INCREASING SEQUENCE *) MINP:=I
38    ELSE (* MONOTONOUSLY NON-DECREASING SEQUENCE *) MAXP:=I;
39    XP1:=XP
40  END;
41  IF MINP>MAXP
42  THEN IF MAXV-MINV>=USLOPE
43    THEN BEGIN IF USLOPE>HOM THEN BEGIN SOM:=SOUSL; EOM:=XM END END
44    ELSE IF MAXV-MINV>HOM THEN BEGIN SOM:=SOUSL; EOM:=XM END;
45  XO:=SOM; XM:=EOM
46  END;

```

Fig. 6. Listing of the procedure that finds the boundaries of one block.

```

0
1
2*
3***
4**
5*
6
7
8****
9*****
0*****
1*****
2*****
3*****
4*****
5*****
6*
7
8
9
0
HISTOGRAM OF X-VALUES.

```

```

0
1
2
3
4
5**
6*
7*****
8*****
9*****
0*****
1**
2*
3
4
5
6
7
8
9
0
HISTOGRAM OF Y-VALUES.

```

Fig. 7. Histogram of dimensions in X- and Y-direction of the objects found by the program.