© von Karman Institute for Fluid Dynamics
Waterloosteenweg 72 – 1640 Sint-Genesisu-Rode

Dear reader,

The electronic document attached is protected by copyright laws.

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden.

Please do not distribute the document to third parties outside your workgroup as under current copyright law it is only intended for your personal scientific use. If you do not comply with the above conditions of use, you may be held liable for copyright infringement.

VON KARMAN INSTITUTE FOR FLUID DYNAMICS

LECTURE SERIES 1984-03

DIGITAL IMAGE PROCESSING IN FLUID DYNAMICS

MARCH 5-9, 1984

POST-PROCESSING : EXTRACTING AND DISPLAYING USEFUL INFORMATION FROM DIGITAL IMAGES

JAVIER JIMENEZ - IBM SCIENTIFIC CENTRE, MADRID, SPAIN

POST-PROCESSING: EXTRACTING AND DISPLAYING USEFUL INFORMATION FROM DIGITAL IMAGES

Javier Jimenez
IBM Scientific Centre
P. Castellana 4, Madrid (1), Spain

Fluid mechanics is an area in which much of the experimental information is obtained in the form of flow visualisation pictures. This is specially true in the area of turbulence, where the flow properties change rapidly in space and time, and where a picture can provide information on the instantaneous behaviour of some variables (notably concentration of different species) over extended regions. Flow visualisation movies are even better in that they show the evolution of those variables in time. Extracting information from this material usually involves two steps. First the images have to be processed to improve their signal to noise content, enhance their desired aspects, etc. and, then, they have to be "understood". Digital image processing has become an important technique in this respect thanks to its ability to provide both relatively simple display, enhancement and measurement capabilities and high level image analysis. These advantages of digital image processing are not limited to fluid mechanics but extend to all the areas in which images are an important part of the result of a experiment.

This course contains many examples of applications of DIP, and many techniques on how flow visualisation pictures can be manipulated and improved. However, pictures are rarely the end product that is sought from an experiment. They give an invaluable qualitative insight on the mechanics of a process but they are difficult to compare to one another or to relate to the quantitative predictions of a theory that has to be tested. Moreover, they are difficult to display and to transmit, and they are expensive to store. Consider that a simple visualisation movie running for one minute at 24 frames/sec and digitised with a resolution of 200x300 pixels, contains almost 90 million pixels. This amount of information would strain the storage capacity of most computers and, even if it could be stored in disk and read into the computer at a rate of 100,000 bytes/sec, it would take almost 15 minutes just to be read.

This is a general problem in any occasion in which more than a few images are being analysed. In those cases it is important to decide early in the process which one is the information that is really needed and to extract it from the images as soon as possible. The original pictures can then be archived for future reference, but the extracted information can be processed much faster and usually much more completely.

A related problem is how to display the information once it has been extracted. There are many techniques available to display numerical and statistical information, but graphical data, especially three dimensional ones, require new methods that are only now becoming available. Imagine for a moment that a magical instrument would give you the time evolution of the instantaneous concentration distribution at every point in homogeneous turbulent mixing. How would you display it, publish it, explain it to somebody or even understand it"

Four dimensional (x-y-z-t) information is still rare in fluid mechanics, but some three dimensional data is beginning to be available, either x-y-z or x-y-t, and the display problems may be formidable. The only other areas in which similar problems have been encountered are art, which has solved them in different ways and with varying levels of success, and medicine which run into the three dimensional display problem with the introduction of computed tomography. We cannot expect yet to approach the perfection of art in the display of spatial information, but we should learn to use the lessons of medicine.

We will discuss here two examples of the extraction and display of information from flow visualisation movies. The two examples are centred on the study of the plane turbulent mixing layer, and were processed at the IBM Scientific Centre at Madrid (1,2) using flow visualisation movies obtained by L.P. Bernal at Caltech (3). The mixing layer is established when two parallel streams of different velocities and composition are allowed to mix at the end of a splitter plate. The flow has been known for some time to be dominated by large, roughly two-dimensional, eddies which are convected downstream at the same time that they grow and occasionally interact among themselves. The purpose in both cases was to study the structure and evolution of these large scale eddies.

In the first example we study a movie showing the spanwise integrated two dimensional structure of the concentration field (figure 1). The general structure of this field was reasonably well understood before the processing of this particular movie was undertaken and the emphasis was in obtaining accurate statistics of the life span and modes of interaction of the eddies. The idea was to isolate individual eddies, trace their histories among frames and compile statistics of the length and interconnections among these histories. A sequence of nearly 400 movie frames was used and special care was taken of keeping the process as automatic as possible so as to minimise the effect of observer bias. A rather classical set of enhancement and classification techniques was used to isolate the mixing region. Then the outline of the region was extracted, and the eddies themselves were identified using pattern recognition techniques. A first approximation to the life histories of the eddies and to their interaction was then formed by tracking the eddies from frame to frame, and these histories were further refined using special techniques to

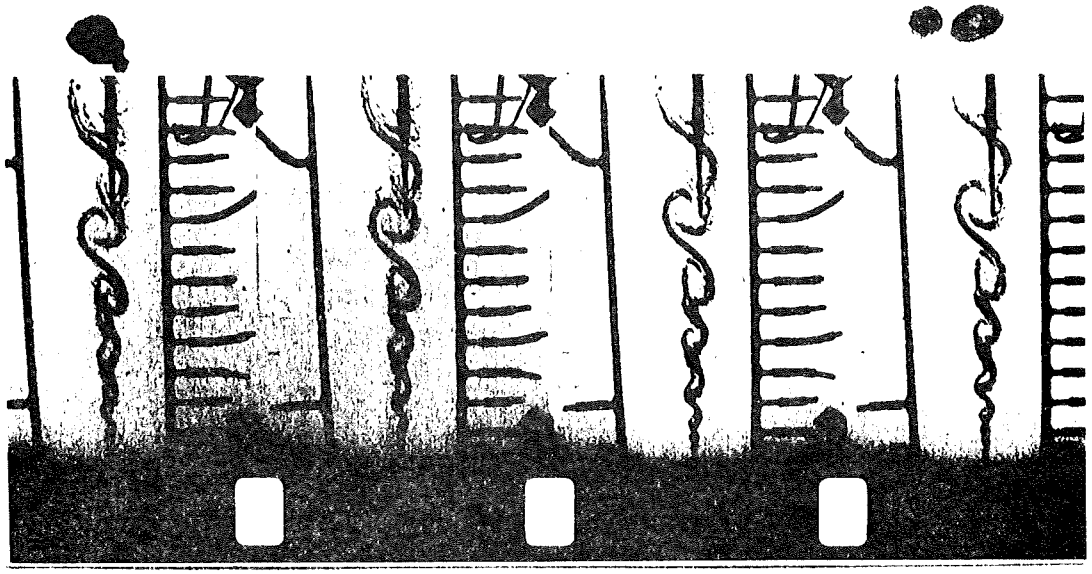


Figure 1.- A few frames from the film used in the first example in the text. Flow is upwards within each frame. Time runs from left to right. (ref. 3)

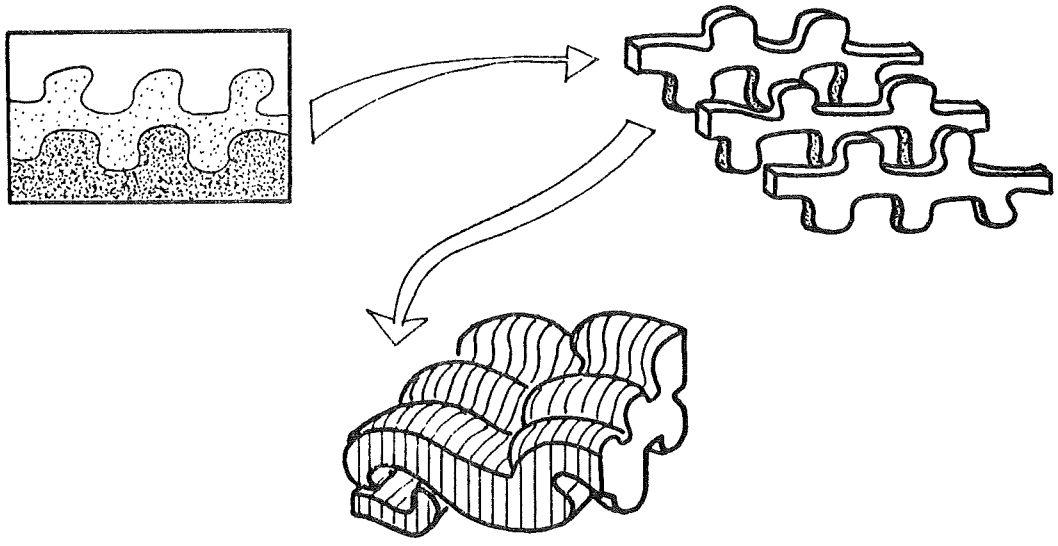


Figure 2.- Reconstruction of a three-dimensional model of the mixing region from sucessive sections. In the model, the flow (time) runs from left to right (ref. 2)

reduce the effect of identification errors in the final statistics (1,4,5).

The reduction to contours was the key step in reducing the amount of stored information. In this step the storage was reduced from the several tens of megabytes needed for the tapes of the original images to 200,000 numbers that could be kept in disk and treated much more carefully. In particular, the reduced volume of information was necessary before the syntactic pattern recognition rules that were used in isolating the individual eddies could be applied. These rules are complicated and iterative, and applying them to the original tapes would have implied an amount of processing time much beyond the capacity of a general purpose computer.

The second example relates to the three dimensional spanwise structure of the mixing layer. The movie used in this case was a laser fluorescence tomography in which the laser sheet illuminated a transverse section of the layer, so that, as time went by, the layer was swept past the sheet by its own average motion (3). As a result the movie could be used to reconstruct an approximate three dimensional (y-z-t) picture of the concentration field in the layer (figure 2). In this case the structure of the field was not as well understood "a priori" as in the previous one, and the emphasis was not so much on accurate statistics as on understanding a complicated geometry through graphical representation. Classical techniques were used again to isolate the mixing region in each frame, and the juxtaposition of all those two dimensional regions was used to construct a three dimensional "solid", representing the layer (2).

The main problem in this case was also one of information compression. Each of the three separate sequences processed, each one containing 200 frames, was equivalent to more than 20 Mbytes after digitisation and had to be reduced to a few hundred Kbytes before it could be drawn interactively. This was done again by isolating the regions of interest and storing them in the form of a contour data base. Several graphic representation methods were then used to highlight different aspects of the structure.

Contours are, in general, a very compact way of storing binary images. Broadly speaking, the amount of information needed to store a two dimensional image grows with the square of the linear dimension, while the number of points in a contour is only linear. The space savings become soon very important. The question is slightly more complicated when the regions to be approximated are very convoluted, which is a typical problem in fluid mechanics. It is clear, for example, that it is possible to imagine a contour so complicated that it would have almost a many points as the area it encloses. This concept can be formalised in terms of the fractal dimension of the contour (6) and it can be shown that the savings in storage are correlated with the fractal dimension of the region (7)

We will now review a few of the key processing steps in both examples. The aim will be not so much to repeat the discussion in the original publications and to present the results, as to illustrate the use of different pattern recognition and information display techniques in different cases.

TWO DIMENSIONAL EXAMPLE

Classification

Contours are extracted from binary images. They are defined as the border between pixels which are "inside" a region and those which are "outside". The distinction between the two regions is generally done using class discrimination techniques, of which the simplest one is thresholding. Of the two examples with which we are concerned here, classification methods are seen best on the first one (figure 3). The first step is filtering to remove the effects of optical vignetting and to decrease the amount of noise. Once filtered, the mixing region is clearly darker than the laminar streams, due to the higher dispersion of light by the density fluctuations. In principle it should be relatively easy to find a threshold which separates both regions but, in practice, it is slightly more complicated because the contrast between the two regions changes downstream, in part due to optical vignetting, but also to the milder density gradients in the downstream stations. The consequence is that a different threshold has to be chosen at different places.

A practical way of doing this, which can be generalised to many other situations is to break the image into local zones and, for each zone, to compute a threshold that locally separates the two regions as efficiently as possible. A global variable threshold can then be interpolated from these values.

In the case of figure 3, the best way is to break the image in vertical strips. Each strip contains both laminar and turbulent regions and the separation threshold is reasonably uniform within it. For each strip, a histogram of pixel intensity would show two modes, one corresponding to light and the other to dark pixels (figure 4). The optimum threshold would be the one that separates best the two "bumps" in the histogram.

There are efficient techniques to find the optimum partition threshold for a bi-modal histogram. Basically they work by making an initial guess, which separates the histogram into two classes. To each class we can fit a probability distribution, usually gaussian, whose parameters are computed from the statistics of the two classes. The point at which both distributions have equal probability can then be used as an improved guess for the threshold, which defines two new classes to which we can again fit gaussian distributions. The process is iterated until the threshold converges. Further

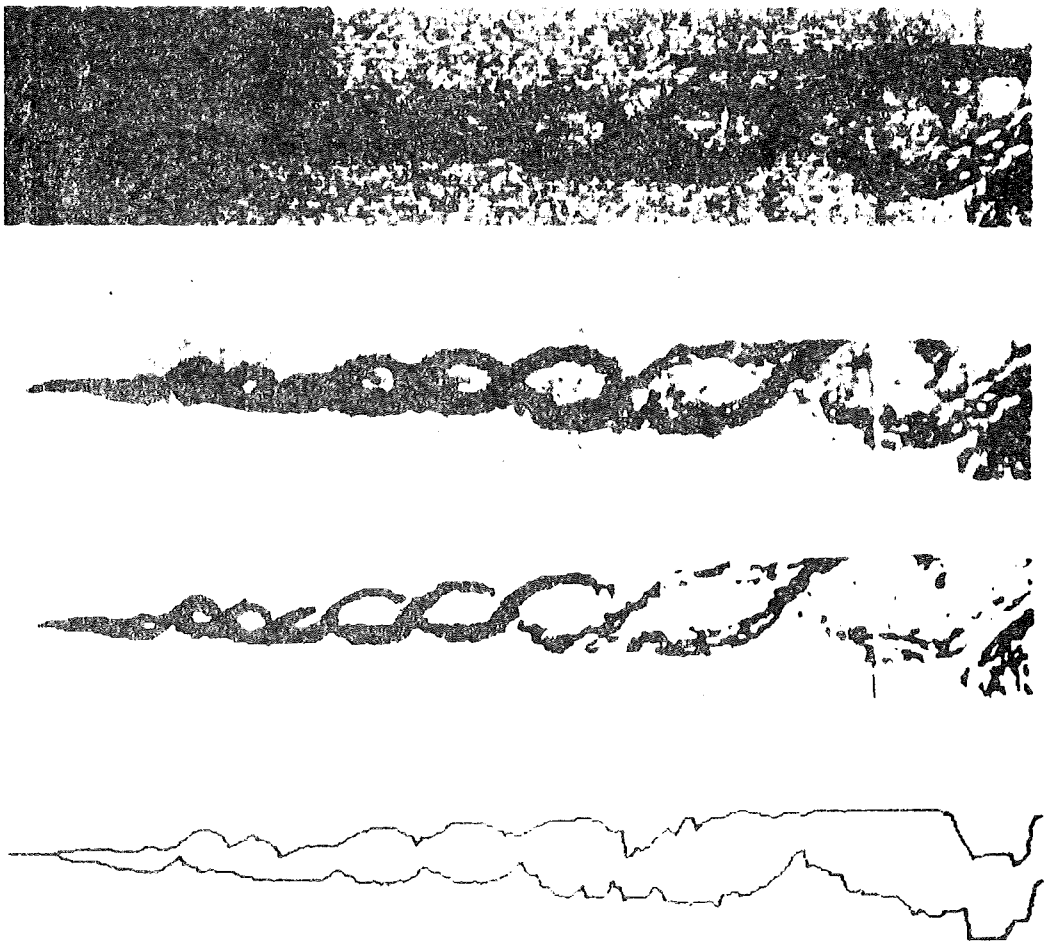


Figure 3.- Preprocessing and contour extraction from one of the frames in figure 1. (ref. 1)

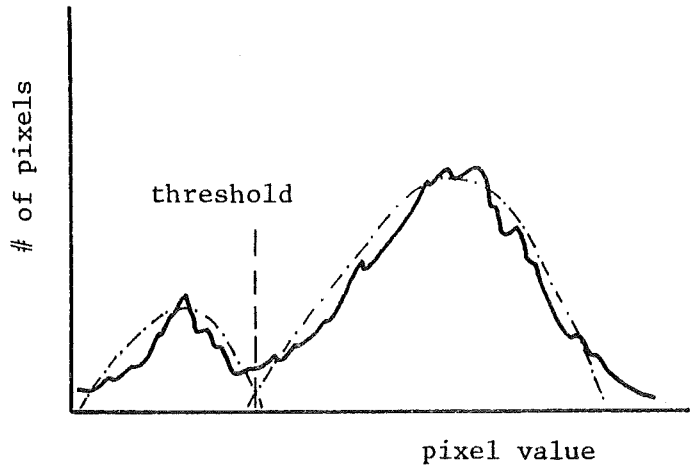


Figure 4.- The histogram of pixel intensity shows two different modes corresponding to differentiated regions. When a gaussian distribution is fitted to each of the "bumps", the optimum threshold is the point where both distributions cross.

details can be found in (8).

Contour following

Once a variable threshold has been determined, the image is classified into two distinct classes, and the next problem is to find the boundary that separates them. There are two different concepts of boundary. A relatively easy task is to find all those pixels which are in the border between the two regions. This generates a new image in which the two regions have disappeared and the only remaining pixels are those in the contour. While this might be useful in some cases, mainly for display purposes, it does not produce any information compression and does not help in understanding the original picture. The new image is just as big as the old one, although probably emptier, and the structure of the region is no clearer than before.

A different problem is to organise all the boundary pixels into one (or several) continuous lines. The end result is a list of points, or a polygon, and, in the process of determining it, it is possible to measure properties like the number of distinct regions, the number and area of individual holes, etc. This operation is called contour following, and is also a relatively straightforward operation. A commonly used method explores in rotational order the eight neighbouring pixels of a point already known to be on the contour and looks for a transition from outside to inside the region (figure 5). In figure 6 we give an algorithm based on this idea. The image is scanned in row order, and the routine contour is entered each time a transition from 0 (outside) to 1 (inside) is detected. From then on, the routine tracks the outline and returns to the scanning program only when a closed

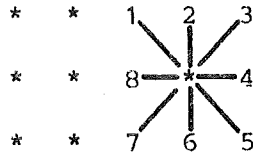


Figure 5.- Pixel neighbourhood and search order in the contour following algorithm.

```
              array image,list,point(2),dir(2);
              input(point); output(list,klis,ext);
contour:      begin;
              first=:true; ext=:false; fin:=false;
              dir=:7;
              klis=:1; list(1)=:point;
search:       do k=:1 to 7;
                 dir=:1+mod(dir,8);
                 if image(point+dir) ≠ 0 then go to keep;
hole:            if first and dir=2 then ext=:true;
mark:            if dir=8 then image(point)=:2;
              end do;
              return;
keep:         point=:point+dir; first=:false;
              if fin and point=list(2) then return;
                                       else fin=:false;
              if point=list(1) then fin=:true;
normal:       dir=:dir+4;
update:       klis=:klis+1; list(klis)=:point;
              go to search;
              end contour;
```

Figure 6.- A contour following algorithm; directions of movement are defined in figure 5 (after ref. 7)

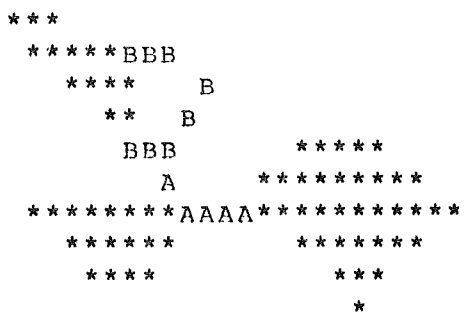


Figure 7.- All pixels marked with an A are traversed more than once in following the external contour. Pixels marked with a B belong to two different contours.

contour has been completed. To simplify the presentation we have ignored complications arising from the object touching the edge of the image, and we have assumed that no buffering is needed and that the whole image is held in memory. Both of these difficulties can be overcome if necessary.

Some features in the algorithm need explanation. The statement normal chooses the current external normal to the contour as the starting point for the next search. This is done based on the knowledge of the direction of the last segment of the contour and ensures that the whole border is followed without accidental shortcuts in cases like the one in figure 7. In the statement mark some of the pixels already recognised as forming part of the contour are marked. This is necessary to avoid re-entering the same contour several times in the course of scanning the image, but it has to be done carefully since some pixels may have to be reused, not only in the same contour, but even in different ones (see figure 7). The solution is to mark with a new code (say, 2) points in the left leading edge of the object, thereby making them unavailable as 0-1 transitions in the scanning while retaining them as 0 to non-0 transitions for the contour follower. The algorithm ends when two consecutive points are found to coincide with the first two points in the contour.

Several refinements can be introduced in the procedure. As an example, it is possible to mark each contour as an external outline or as a "hole" by analysing the search around its first point (this is done in the statement hole in figure 6), thus saving considerable processing in some later operations. It is also possible to do a zero error polygonal approximation, while the contour is being generated, by keeping only those points in which the direction changes. This saves storage.

The bottom part of figure 3 is the result of extracting the contour of the mixing region that was isolated in the classification step. To further simplify the representation, all the areas smaller than a certain size have been erased, as well as all the interior holes. Furthermore, re-entrant sections of the contour have been eliminated and the resulting line has been smoothed considerably. The result is a connected outline that contains all the essential information for the recognition of the eddies but which can be stored in much less space.

Segmentation

The key step in many problems of image understanding is the identification and isolation of the individual elements which are meaningful from the point of view of the final application. Once these parts are isolated many of the activities of measuring, locating and integrating them into more general patterns are relatively easy. In the particular case of our mixing layer, the goal was to identify individual eddies so that they could be tracked as they moved in time,

and their behaviour be measured. Unfortunately the shape of the eddies is variable and ill-defined and the identification of a given part of the contour as an eddy can only be made in a statistical sense. The aim is then to segment the mixing layer in such a way that the probability of the segmentation be maximum.

This is done in three stages. First the contour of the mixing layer in broken into streamwise segments in terms of "bulges", local fattenings that are likely candidates for eddies; next these segments are classified as eddies or braids according to their geometric properties, and finally an attempt is made to improve this classification by merging neighbouring segments and checking the properties of the resulting unions. Only the first two steps will be discussed here. Further details can be found in (1,5)

The description of a "bulge" is done best using syntactic pattern recognition methods. The layer has first to be broken (parsed) into elemental segments. These segments have no meaning in themselves, but they are the building blocks from which our meaningful units will be built. It is useful to consider the elemental segments as letters, and the final units as words. Then, the description of which ones are the words in which we are interested takes the form of a grammar containing rules about which letters, and in which order, can be catenated into syllables or words, and which ones cannot. The study of these pattern recognition methods has attained a high level of formal rigour; a survey can be found in (9).

The properties used for the elemental parsing of the layer are the slopes of its upper and lower edges, and the elemental segments are bounded by the maxima and minima of those edges. A segment is classified as divergent -A-, parallel -B- or convergent -C- according to which is the slope of its edges and to the absolute magnitude of some properly defined measure of divergence (see figure 8). A frame is then written as a sequence of the form -ABAC...CCAB- and a set of grammatical rules is established whose aim is to simplify the sequence and to isolate units either of the form -AC- which are the bulges, or of the form -...C-B-A...- which represent the intervening necks. The basic idea of the rules is that convex corners in the middle of a divergence or convergence are erased while concave ones, or changes from divergence to convergence, are preserved as final segmentation end points. It can be shown that, by applying these operations repeatedly, a state is reached in which the layer contains only bulges, -AC-, or necks, -B-, and, moreover, that this decomposition is unique. An example of simplification and segmentation is given in figure 9.

These bulges and necks are not necessarily eddies or braids, but they constitute a first segmentation of the layer into individual pieces that can then be classified according to their shapes. Assume that we characterise this shape by the maximum width and the length of a segment. If we had

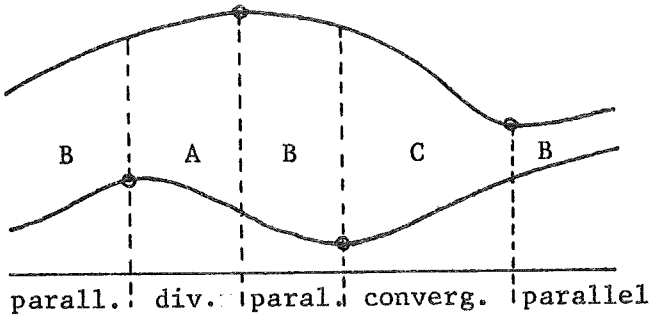


Figure 8.- Definition of the elemental segments in terms of convergent or divergent sections of the mixing region. The ratio between divergence and length is also used in the classification.

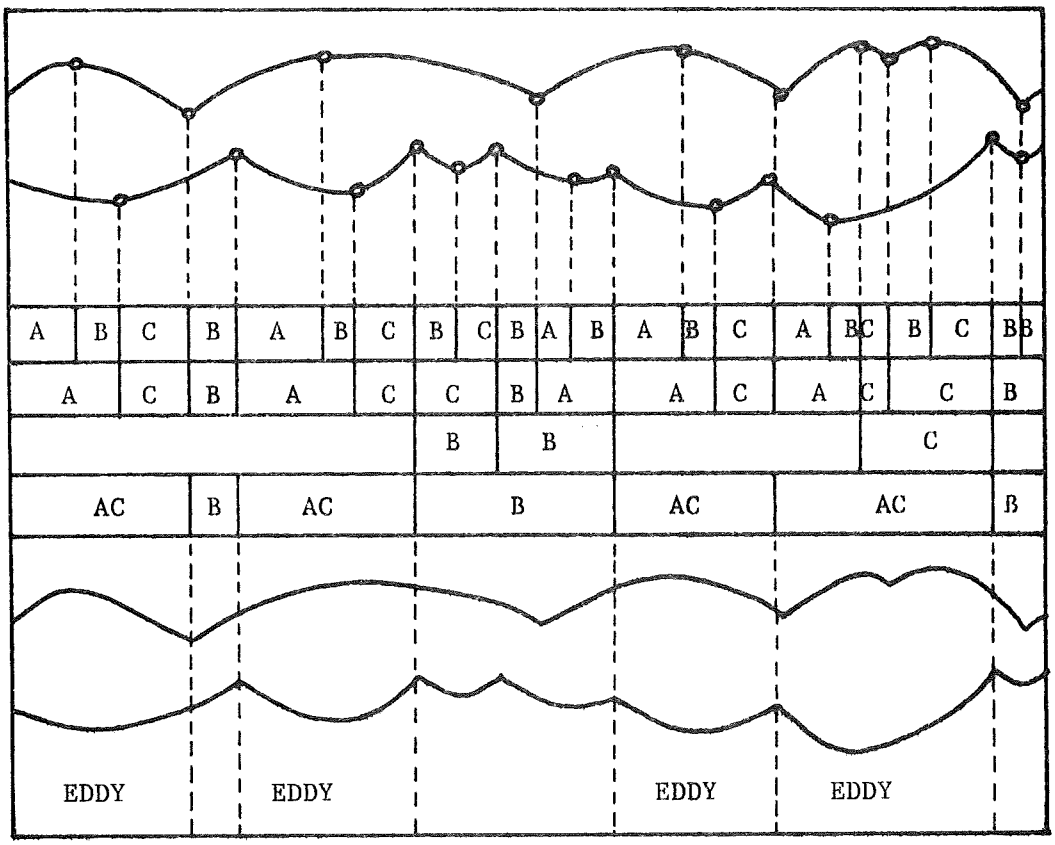


Figure 9.- An example of the syntactic simplification process. Four steps are needed to go from the initial segmentation (top) to the simplified string at the bottom (after 4).

statistical information on the bi-variate distribution of these two quantities for eddies and braids, we could classify each segment as one or the other using those distributions. This information is acquired by training the computer on a set of known cases.

We choose a subset of a few frames, representing a small percentage of the total, and subject it to the processing described up to now, including the first segmentation. These frames and their segmentation are then displayed in an interactive graphic terminal where each segment is classified either as an eddy or as a braid by an experienced operator. In this way we get a controlled sample of about one hundred eddies and roughly as many braids, together with their dimensions, that can be used to build the desired statistics (figure 10).

The distribution of shapes for the two kinds of segments clusters into two clouds whose underlying distributions can be approximated as gaussians so that, when a new segment of unknown type is found, we can, using these distributions, not only classify it as an eddy or as a braid, but assign to it a probability of belonging to each class.

This is an example of classification which differs from the one presented for the thresholding of the original images. There, the algorithm itself was given the task of generating the two probability distributions for light and dark pixels. Here it is the operator who makes the initial training and decides how the different patterns are classified. Algorithms of the first kind are usually referred to as unsupervised classification or "clustering", while those in the second are called supervised.

Both types of algorithms have different applications. Unsupervised classification works best in simple cases in which many patterns have to be classified on the basis of a few characteristics. In these cases the "brute force" approach of clustering methods is usually able to distinguish among different groups, if they exist. In those other problems in which the number of phenomena to be classified is small or in which the number of variables used for discrimination is large (in the present case, 2), clustering methods fail and the classifier has to be guided by the operator.

The reason is clear. Consider an individual pattern (object) as a point defined by N coordinates. An example is a pixel characterised by its gray level (N=1), or a mixing layer segment defined by its length and width (N=2). When a set of these objects is to be partitioned in several classes, the goal of the classifier is to find regions in N-dimensional space in which a point has a higher probability of belonging to a given class. Clustering algorithms work by dividing the space into "a priori" boxes and checking whether any of those boxes are more densely populated than others. It then assumes that those regions correspond to individual classes and

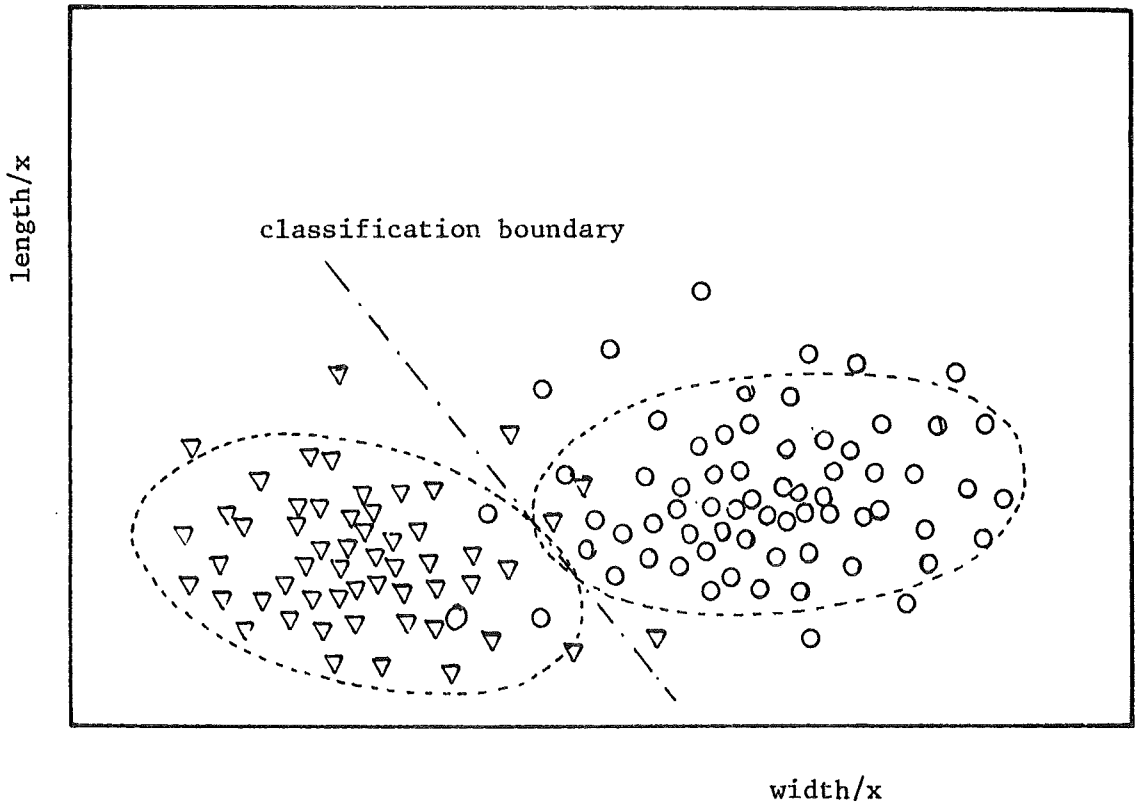


Figure 10.- Statistical distribution of shape parameters for different types of segments. (o)- eddies; (▽)- braids.

classifies neighbouring points according as to whether they are closer or further away from any of those accumulation centres. It is clear that the higher the number of points available for testing, the easier it would be to find a box with a clear accumulation of points. If only a few patterns are available, their random distribution might be too sparse for recognition, or even very different from the underlying statistics. The problem is that the higher the dimensionality of the space in which we have to search, the larger the volume on which the different patterns can be distributed and the sparser their distribution will be. In practice it is difficult to use unsupervised classification in problems in which the number of characteristics involved is higher than two or three. Beyond that, the initial classification has to be done by the operator and the computer is only used to compare the rest of the patterns with the models provided.

In the example that we are treating here, the number of total objects, a few hundred bulges and necks in a two dimensional space, is too low for a clustering algorithm. Supervised classification could, however, be used successfully. An example of the result of the segmentation of a few frames, followed by tracking among frames and by the construction of histories is given in figure 11.

Classification methods based on the comparison of numerical patterns with probability distributions are known as statistical pattern recognition, and they are fundamentally different from the syntactic methods used for the initial segmentation of the mixing layer and which are based on relational "logical" rules. The example that we have presented here shows that both systems are not mutually exclusive and that they actually become most powerful when used together. A general reference book on statistical pattern recognition is (10).

TRANSVERSE SECTIONS

Graphics

As explained before, the problem with the second movie of the mixing layer is one of representation. The information extraction steps are very similar to those of the first one. Each frame is filtered to remove noise and shading and classified. The classification, in this case, has to define three regions, each corresponding to a different range of dye concentration. In the visualisation method, a dye is introduced in one of the free streams and is used as a tracer for the mixing. Therefore, one stream is free of dye, the other one has unit concentration, and the mixing region in between has a concentration intermediate to those two extremes. Therefore, the classification step needs two thresholds, instead of one. In this case, the thresholds were defined manually for one image and all the other images were radiometrically corrected so that the same values would hold. The classification of one frame into its three regions is

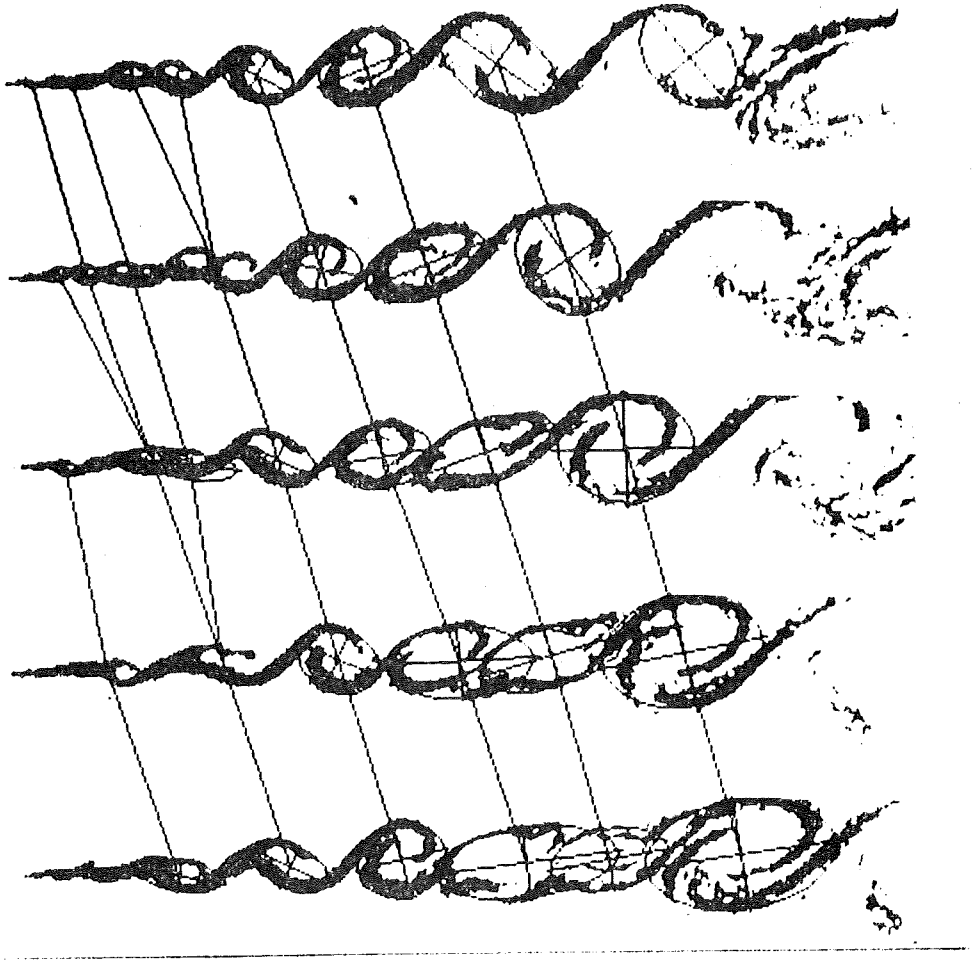


Figure 11.- Part of the x-t diagram of eddy histories after the layer has been segmented and the eddies tracked between frames. (ref. 1)

shown in figure 12. The mixing zone is the gray area in the middle; the other two areas are the free streams. The splitter plate is horizontal, and the flow is out of the page towards the reader.

At this stage the information is still in the form of a large tape containing several Mbytes of image information. Before anything else can be done, the important information has to be extracted. What we are interested in this case is in the shape of the mixing region which is, of course, contained in its outline. Those outlines are extracted using exactly the same procedure described above and stored in a disk file. Each outline, which is normally formed by several independent polygons and holes, represents a slice of the mixing layer and, when all of them are juxtaposed together, they form a three dimensional solid that approximately represents the spatial shape of the mixing layer (the relationship between the y-z-t distribution actually observed and the x-y-z instantaneous concentration field is discussed in reference 2).

The reconstruction of the three dimensional body is conceptually equivalent to pasting each contour onto a piece of plywood, cutting them out, and stacking the pieces together (figure 2) In fact, this procedure is still used at times by biologists in reconstructing structures from microscopic sections, and at least one machine is commercially available that will carve a block of plastic foam starting from the file of stacked contours. A more efficient procedure is to use computer graphics to synthesise different views of the object. There are several commercial graphic packages that can use a plotter to draw a surface given as an elevation matrix or in the form of a one-valued function of x and y. These packages would not usually take shapes as complex as the ones produced by the contours of the regions in figure 12. The general problem is to draw a set of polygons in space while erasing the hidden lines.

Hidden line removal is a difficult process in which the key problem is one of ordering the sides of the polygons in such a way that as few possible intersections as possible have to be tried. When the polygons to be tested are such that there are no points with vertical tangent (except the two end points), the ordering is trivial (say, left to right); this corresponds to most of the packages normally available. When return points are present, the problem becomes more involved and is outside the scope of these notes. An elementary account of the general case of hidden line removal can be found in (11). The best sources for up to date information are the specialised journals (12).

Two examples of display using simple stacking of contours plus hidden line removal are given in figures 13 and 14. The first one is a set of contours extracted from the film of transverse mixing layer sections. The resulting body shows the three-dimensional shape of the mixing region and has been

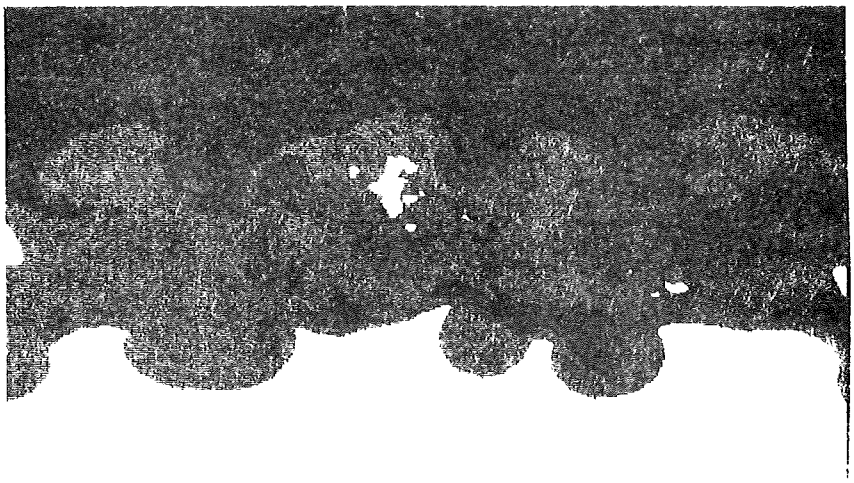


Figure 12.- One frame showing the tranversal section of the mixing layer, once filtered and classified. Flow is towards the reader

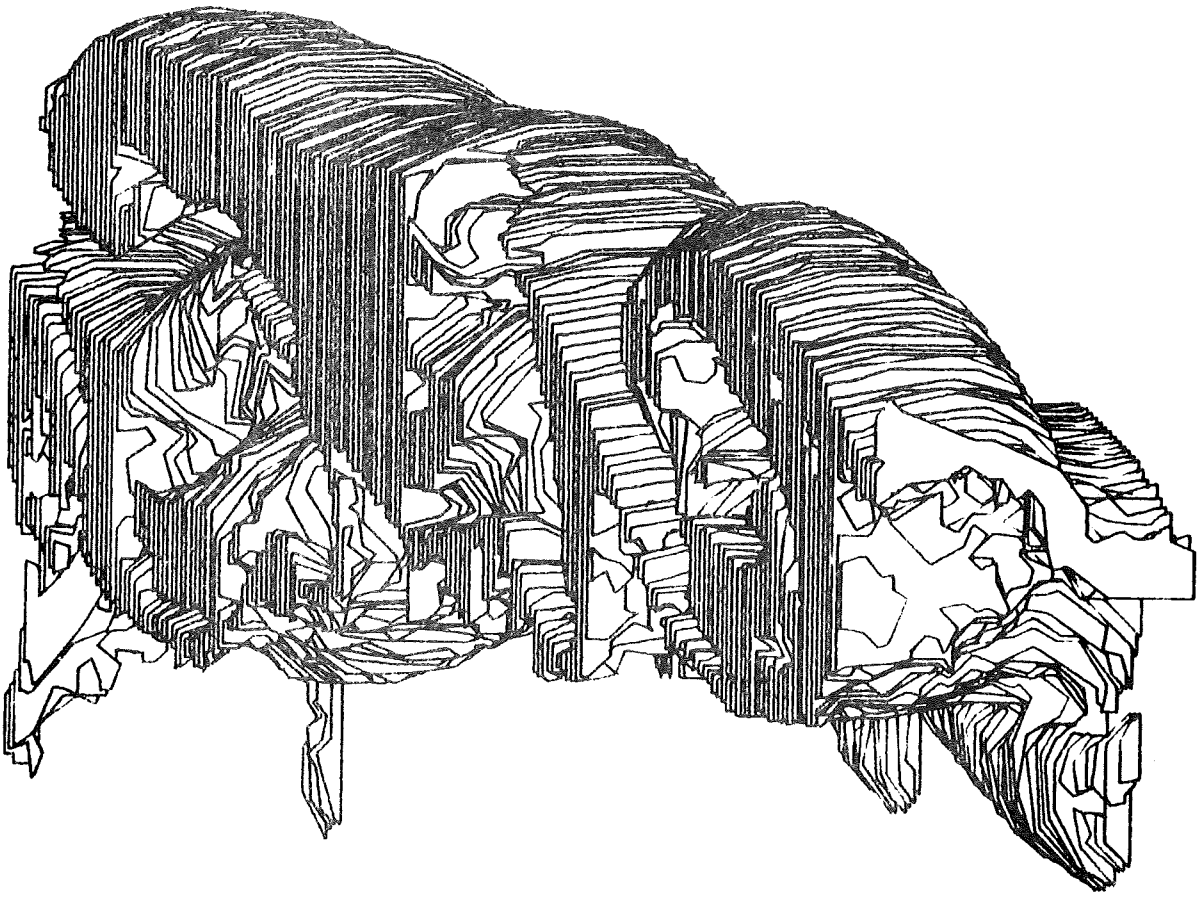


Figure 13.- Model resulting from stacking the contours from several sections of the mixing layer. Flow is top left to bottom right.

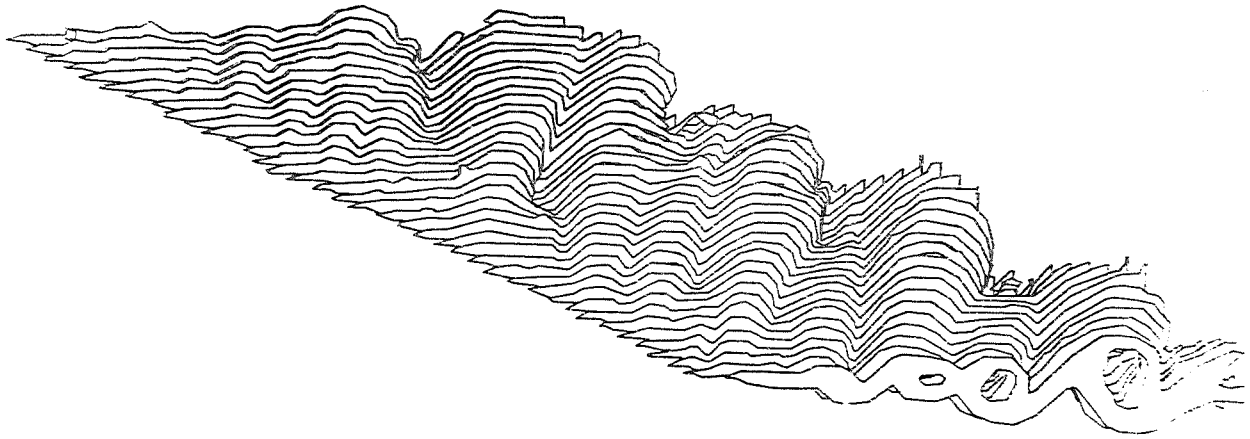


Figure 14.- The evolution in time of the two-dimensional eddies can be displayed by stacking the outline of the mixing region from several frames of the film in figure 1. The layer is seen from the high speed side. Flow from left to right.

cut by a streamwise plane to show the internal structure. The example in the second figure is a stacking of the contours in the two dimensional mixing layer movie. While this representation adds little to the quantitative description of vortex movement and amalgamation, it does give a good qualitative global feeling of the processes involved.

There is a hidden line algorithm which is especially easy to implement, although it uses large amounts of memory and implies the use of a raster output terminal. Assume that we have our sections in the form of binary images. The inside is marked as one and the outside as zero. This is, in fact, the representation that we had before the contour following step. Note, however, that, because secondary storage is so much slower than core, it is generally faster to store the contours as polygons and to reconstruct the binary representation each time, than to store and read the binary images from tape. There are efficient algorithms for polygon filling; a good survey, which also contains many other useful algorithms, is (13).

Define two buffers of the same size as your display screen. One buffer will contain the final display and is initialised to zero, while the other one is the depth buffer and is initialised to a large negative number. The stack is drawn beginning from the polygon farthest away from the observer. All the pixels which fall inside the polygon are substituted in the display buffer by whatever colour is desired, while in the depth buffer they are made equal to the (negative) distance from the observer to the polygon. When the next polygons are drawn, each point is first checked against the depth buffer; if the new point is nearer than the distance written in that point of the buffer, both buffers are updated; if it is farther away, the new point is hidden and it is discarded. The algorithm is actually faster in most cases than polygon intersection methods, but it uses a lot of memory and it is only able to produce solid representation of faces (not outlines) in display devices in which each pixel is assigned a colour (raster monitors and hard copies). It is most effective when used in representing polyhedral models in space, in conjunction with shading or colouring of the different faces. A classic reference for shading is (14) and a good survey of raster graphics is again found in (11).

Two examples of solid modelling using shaded polyhedral models and depth buffer hidden lines are given in figure 15. They both correspond to stacks of transversal sections of the mixing layer, which have been turned in polyhedrons by simply giving to each section a finite thickness. Each representation takes two or three minutes in a medium size mainframe computer (an IBM 370/158), and can be manipulated interactively, both in orientation and illumination, or sectioned through any



Figure 15.- Two different perspective views of the mixing layer three-dimensional model obtained by stacking transverse sections. In both cases the flow is roughly left to right.

chosen plane.

CONCLUSIONS

In conclusion, we have given a few examples which show that image processing, together with techniques in pattern recognition and computer graphics, can be used profitably in analysing large amounts of experimental data in fluid mechanics. This is true not only of cases like the first one, in which the statistics could have been obtained manually, although through a extremely tedious process open to errors and unconscious operator bias, but in other cases, like the second example, in which there are virtually no other ways of obtaining the same result.

REFERENCES

1.- M.A.Hernan & J.Jimenez, "Computer analysis of a high speed film of the plane turbulent mixing layer". J. Fluid Mech., 119, pp. 323-345 (1982).

2.- J.Jimenez, M.Cogollos & L.P.Bernal, "A perspective view of the plane mixing layer" Proc. 4th Symp. on Turbulent Shear Flows, Karlsruhe, (Sept. 1983), pp. 16.7-16.12

3.- L.P. Bernal, "The coherent structure of the plane turbulent mixing layer". PhD Thesis, California Institute of Technology, Pasadena (1981).

4.- M.A.Hernan & J.Jimenez, "Image analysis of experimental data in turbulence". Proc. 8th IFIP Congress, Melbourne, (Oct. 1980), pp.753-756

5.- M.A.Hernan & J.Jimenez, "Statistical Segmentation of Images" Proc. Int. Conf. on Digital Signal Processing. Florence, Italy, (Sept. 1981), pp.985-992

6.-B.Mandelbrot, "Fractals: Form, chance and dimension", Freeman, S.Francisco (1977)

7.- J.Jimenez & J.L.Navalon, "Some experiments in image vectorisation" IBM Journal Res. Dev., 26, pp. 724-734 (1982).

8.- C.K.Chow & T.Kaneko, "Boundary detection of radiographic images by a threshold method". In Frontiers in Pattern Recognition (ed. S.Watanabe), pp. 61-82. Academic, N.Y. (1972)

9.- K.S.Fu, "Syntactic methods in pattern recognition", Academic, N.Y. (1974)

10.- R.O.Duda & P.Hart, "Pattern classification and scene analysis", Wiley-Interscience, N.Y. (1973)

11.- J.D.Foley & A.Van Dam, "Fundamentals of interactive computer graphics", Addison-Wesley, Reading, Mass. (1982)

12.- A good source are the publications of the ACM special interest group on graphics (SIGGRAPH), especially their quarterly review "Computer Graphics". One issue contains the Proc. of the annual SIGGRAPH Conference. The address of the Association for Computing Machinery is 11W 42nd St.,New York, N.Y. 10036.

13.- T.Pavlidis, "Algorithms for graphics and image processing", Springer-Verlag, Berlin (1982)

14.- B.T.Phong, "Illumination for computer generated pictures" Comm. ACM, 18, pp.311-317, (1975)