# Automatic Pattern Recognition

by R. J. Petheram, BSc.

Thesis submitted to the University of Nottingham for the degree of
Doctor of Philosophy, October 1989

# Contents

# Abstract

In this thesis the author presents a new method for the location, extraction and normalisation of discrete objects found in digital images. The extraction is by means of sub-pixcel contour following around the object. The normalisation obtains and removes the information concerning size, orientation and location of the object within an image. Analyses of the results are carried out to determine the confidence in recognition of patterns, and methods of cross correlation of object descriptions using Fourier transforms are demonstrated.

## Acknowledgements

My grateful thanks go to Professor R.L.Beurle for all his help and encouragement over the years of this project, his advice and guidance have been invaluable. Gratitude is also due to R.S.R.E Great Malvern for funding the research project, and for the helpful comments that Mr. J.Sabey in particular has provided at contract review meetings. My appreciation also goes to my wife Louise who typed and proof-read the manuscript. Thanks also to Mr. Farhang Daemi for his contributions over the years of the project and his help with the final preparation, and to my parents for their support and encouragememnt.

Richard Petheram

# Introduction

The work described in this thesis has arisen from a research project sponsored by R.S.R.E. Great Malvern. The broad subject of the research is "Automatic Pattern Recognition". The research has developed from a long association with R.S.R.E. during which time the subjects of human perception and pattern recognition have been studied. The direction of research, however, has altered during the past few years to be directed away from human perception, towards that of machine perception and recognition.

The area of research as defined by R.S.R.E. has been very wide. The research has been largely directed to recognition of visual images rather than other forms of input, partly for historical reasons, and partly to complement other projects. The author has chosen to limit his own area of research simply to allow a set goal to be defined, thereby giving a direction to the study.

Two areas of research have been investigated by the author in the three year research period.

The first year was spent investigating a position independent method of recognition using Fourier and Walsh transforms. The test data used were based upon infra-red images supplied by R.S.R.E. The results show a limited degree of success in extracting the location information for a single object in an image leaving the detail intact. The work is described in chapter 3.

This led to a realisation of the importance of discarding the vast amount of irrelevant information in a visual scene in order to be able to extract the relatively small amount of information directly relevant to the object to be recognised. Accordingly the second and third years were spent in producing a recognition method based upon the shape information of an object in an image. The thesis describes the method used to construct a set of test data, the approach used in extracting the shape information in terms of a contour around the object, and several approaches to the recognition of the contours produced. Use is made of the work done on Fourier transforms in extracting a normalised description. This work is described in chapters 4 and 5.

# General

## 2.1 The Pattern Recognition Task

### 2.1.1 The reasons for pattern recognition.

The basic reasons for most current Automatic Pattern Recognition tasks are to remove the need for a trained operator to perform the recognition, or to enable recognition tasks to be performed that would otherwise be impossible. The method of recognition used is determined by the task to be performed, and by the cost of implementation. Methods vary from simple very fast routines for real time evaluation of an image, to comparatively slow processes used to automate otherwise tedious tasks.

To solve these Pattern Recognition problems research has been funded, both by public and private sector concerns. The nature of research undertaken is directed by the nature of problems to be solved. Many researchers have entered the field of Artificial Intelligence, and it can be argued that the fields of "Pattern Recognition" and "Artificial Intelligence" are one and the same; certainly there is much overlap. It should be noted, however, that pattern recognition includes the use of very simple template matching which would not in any way class as intelligence, but may nevertheless solve a pattern recognition problem. Conversely, Artificial Intelligence problems would include studies of solutions to problems of philosophy and machine self-awareness which would far exceed the scope of Pattern Recognition. This thesis is principally concerned with pattern recognition, particularly with pre-processing algorithms and has thus not touched on Artificial Intelligence.

To understand the process of pattern recognition a great deal of research has been undertaken covering a wide variety of disciplines, all aimed ultimately at solving the problem of extracting recognisable information from the real world, or solving some part of that problem.

Pattern recognition research has also attracted much interest in Universities where research has not always been tied to the solution of immediate practical problems. This has left workers free for more speculative research addressing problems that do not yield a complete solution to pattern recognition, but which may add to the knowledge of the overall field, thus paving the way for further advances. It is the belief of the author that this speculative research is as likely to lead to tangible results as any other form of research.

The task of a Pattern Recognition machine is to extract specific information (the required detail) from a set of input data. In general the required detail is represented by patterns within the input data. It is usual for much of the input data to provide none of the required detail, and for many alternative patterns to validly represent the same part of the required detail.

In practical terms the pattern recognition task is often one of identification of a specific object (or one object from a set), or to determine whether an object conforms to a predetermined shape within pre-specified tolerances.

The input data is frequently based upon digitised visual images, but can also be taken from other transducers that may be constructed to suit specific tasks.

### 2.1.2 Some parameters of the ideal pattern recognition machine.

In order to assess the performance of a pattern recognition machine during research it is important that the task be well defined. In the most generalised case the required detail is not known prior to the image being seen and anything "interesting" is sought. This is unsatisfactory for an investigation of the principles of pattern recognition since it does not readily allow for the research to be refined on the basis of previous results. For this reason it is more usual for a closely defined task to be investigated, and the results of other research to be used if such use would be beneficial.

Pattern recognition as a whole is too large a field for this thesis, thus the author has chosen to limit the study to a small subsection of the topic. The scope of the thesis is described fully in section 2.4, however it should be noted here that the research has been principally concerned with algorithms for shape identification.

The following is not an exhaustive list of qualities that the ideal shape identification algorithm should possess, however it may be used as a guide for assessing the performance of any particular solution in the absence of any other guide.

- ☐ The recognition process should be independent as far as is possible from the transfer characteristics of the input transducer (eg. the scanning grid in digital image recognition).

- ☐ The algorithm should be independent of the orientation of any object to be recognised, except where the orientation gives information that aids recognition.

- ☐ The algorithm should present information suitable for further processing. Those parameters such as size and location that can be represented by a small range of values and which are sometimes irrelevant to recognition should be extracted and held independently from other information.

- ☐ The algorithm should concentrate data points at regions of greatest "busyness", where the rate of change of value over a small region is greatest, but should allow for the resolution to be adjusted. It should be noted however, that noise can cause regions to appear busy without containing any of the required detail, thus any a-priori knowledge about the noise in the image should be used.

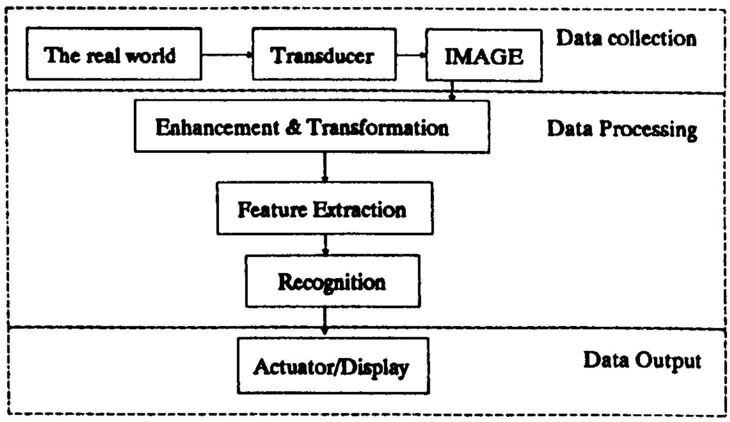
- ☐ The number of data items extracted in any particular class (e.g. contour description) should be determined by the algorithm, and not by the quantity of data input.

- ☐ The algorithm should be constructed to reduce or eliminate noise, and should not add any significant noise of itself. In particular any significant quantisation noise should be avoided since this is entirely under the control of the designer.

### 2.1.3 Functions of the pattern extraction algorithm

The required form and accuracy of the result, when taken with financial considerations, determines the nature of the approach taken to analysis and also to the information gathering transducer. Since there are many different problems that can be grouped into the topic of pattern recognition, so there are many different approaches that may be taken to the subject and no single solution is appropriate to all problems.

The process of pattern recognition is characterised by three operations. These are the data collection, data processing and data output. see figure 2.1.




2.1 Effect of local averaging on a step function

#### 2.1.3.1 Data Collection

In the data collection phase a series of electrical and mechanical devices are used to collect information about a scene, and convert this into an image for subsequent processing. Some preprocessing may be performed during this process, either to minimise errors in data collection, or to reduce the quantity of data that must be stored or transmitted.

The problem to be solved determines the type of data collection required. Many types of sensor have been used, varying from a single inexpensive light sensor, to an infra-red CCTV device launched into orbit around the earth by a rocket. Almost all of the more recently documented research uses analog sensing devices with multi-value analog to digital conversion as the next stage in the recognition process.

The most common type of input transducer used is the CCTV device, with an analog to digital converter which changes the output to a matrix of data values. Commercial digitisers give arrays of up to 1500 by 1500 pixcels at 50 frames per second. Other input transducers use audio, infra-red, ultra-violet and vibration sensors to provide data values. It is rare for there to be more than a small number of discrete sensors (about 6) in an array other than in the case of video input, however the rate of data output can be large, and a single digitised microphone output may give a bandwidth of some tens of kilohertz.

In some instances the image collection and processing are performed simultaneously, often in "real time" (i.e. at the speed determined by the need for results). This is done where the penalties for a late result are high, for example target acquisition for gun aiming in military applications.

Similarly if the problem is dependent upon a decision before any action may be taken then real time analysis must also be used. This is applicable to inspection of goods prior to packing, and is used by exporters of soft fruits, and is also applicable to automated assembly.

Other practical systems use very specialised transducers to collect and digitise images, which are then transferred to a mass storage medium for later and more convenient analysis.

Examples of this may be found in satellite photography where the time of photograph is bound by the location of the satellite, and in medical imaging, where the patient may need to be scanned at one time, and a consultant supervise the analysis at another.

#### 2.1.3.2 Pattern extraction

The first part of the pattern extraction phase is that of image enhancement. This is the process of increasing the signal to noise ratio of the image by removing some of the noise. There are many approaches to this problem, however all of them involve some a-priori knowledge of the image, and may use some form of data normalisation to distribute the data more evenly over the possible range of representation.

In the data processing phase, pattern analysis algorithms follow a set of prescribed rules to deduce the required detail from images. The detail that is sought may be divided into two kinds. Firstly there is the global information that contains the required detail. By careful analysis it is hoped that methods may be developed to determine which parts of the input data may contain the required detail, and which parts do not. It is expected that this study will result in computer algorithms to automate the extraction of the required detail.

Secondly there is the application of the pattern extraction algorithm to images. This is will extract the required detail that a pattern recognition machine is looking for.

#### 2.1.3.3 Data Output

The data output phase is simply a report on the outcome of analysis, however it is as important as each of the other parts of the pattern extraction process, since it is the information that the user actually wants, either as a result in itself, or for further analysis and correlation with other pattern analysis results. The way in which the results are produced and the method of presentation must therefore be carefully considered.

### 2.1.4 Areas of research

The types of study that are currently being researched are many, however the following list covers the majority of research problems that are being undertaken.

- Time consuming, but relatively simple interpretation of information, for example in blood count tests for hospitals. The importance of machine recognition is not in the speed, but in accuracy of result, and in not requiring rests or becoming bored with the problem.

- Total automation of sensing and control of machinery by virtue of dangerous location such as in a battle zone. The difficulty of the research task in this instance is very great, however the funding for such research is also much less constrained than in the commercial sector, and more speculative research is undertaken for military than for commercial ends.

- In circumstances where a human operator would interfere with the sensing. This is true in instances of military sensing where an operator would be endangered, and could possibly be detected. An automated recognition machine may be sufficient to provide useful information, such as determining the presence of a heavy vehicle on a road by sound without being confused by the similar sound of a heavy fall of rain etc.

- Data compression as used in satellite transmissions or for high definition television signals. The use of pattern recognition techniques to code and decode signals where bandwidth is limited is a long established field of research.

- Automated assembly and inspection where simple guidance of a machine may be achieved in a controlled environment, and inspection of machined parts in a similar environment can be achieved.

- ☐ Where there is no simple algorithm one may try many tests and chose the best. Such tasks as determining the molecular composition of a chemical from a gas chromatogram, or the illness of a patient from a number of specific symptoms.

## 2.1.5 Classes of problem encountered

There are many problems faced by researchers in all aspects of pattern recognition that have not thus far been mentioned in this thesis. This list enumerates the problems most commonly tackled.

### 2.1.5.1 Indeterminate and unique patterns

Indeterminate and unique patterns eg. faces or trees, present problems in defining what pattern is being sought. It is not easy to define the features by which a person is recognized even though this problem is readily solved by human beings.

### 2.1.5.2 Occlusions and incomplete patterns

Occlusions and incomplete patterns present problems of the multiplicity of potential solutions. Attempted solutions to this problem are to extrapolate the missing detail or to attempt recognition of the parts and then reconstruct the pattern entity syntactically.

#### 2.1.5.3 Three dimensional data in a two dimensional scene

This is a common problem whereby a 3-dimensional scene is viewed using a conventional camera. The depth information is lost for all objects in the scene, thus the distance of an object from the camera can then only be determined by using foreknowledge, eg regard to size, in interpreting the image.

Also the apparent shape of an object is dependent both upon its orientation relative to the camera and on the effects of perspective due to its distance from the camera. For this reason the number of different pattern entities known about by a recognition algorithm may need to be very large, even to recognise a single object correctly.

#### 2.1.5.4 Moving objects

Information regarding the motion of an object may be crucial to the recognition of the object. Multiple frames must be correlated and the registration of moving patterns must be performed between frames. This information must then be turned into vectors representing the translation and rotation of an object in three dimensions.

#### 2.1.5.5 Variation in lighting

In many instances it is not possible to control the way a particular object for recognition is perceived. Variation in lighting conditions is such an example, and algorithms that are heavily dependent upon texture information for recognition will be particularly prone to problems. The most often used solution is to normalise for intensity of lighting by pre-processing the image (eg. statistically calculate a threshold for segmentation, see section 2.2.2.1 on Thresholding by Mode Splitting).

#### 2.1.5.6 Noise in collection

Noise that is due to the input transducer is a problem that has no solution, except to purchase better image gathering equipment. It is possible to remove some noise by use of appropriate filters, however if the bandwidth of a signal is limited on input there is no amount of filtering that will improve the quantity of information available to the recognition algorithm.

#### 2.1.5.7 Quantisation noise versus speed of calculation.

A balance must be struck between the precision of representation of data and the speed of calculations performed on that data. The usual approach to this problem is to represent the data to a precision that gives satisfactory results, and then to attempt to increase the speed of calculation by appropriate modification of the hardware or software. It may, however, be acceptable to reduce the precision of calculation to speed up the algorithm. This is particularly true in cases where integer arithmetic can be used throughout a calculation because an integer result is required. To use a high precision representation in such cases will not necessarily improve the result, but may reduce the speed of processing by orders of magnitude.

## 2.2 Standard image processing techniques.

A number of techniques for enhancing digital images, extracting edges and grouping patterns into sets are documented in several text books on pattern recognition, and are used as the basis of further research by workers in the pattern recognition field. A brief survey of the relevant methods is presented here.

### 2.2.1 Transformations

Frequently the first requirement of a recognition process is to clean up the entire digitised image, or to enhance particular classes of features in the image. Such operations are applied globally to the image without regard to image detail, and are independent of the position of any particular pixcel within the matrix. In several instances (e.g. Fourier transforms) all of the pixcels in the input matrix are used to find the output value for each individual cell.

The process of such global alteration is one of transformation. The Nearest Neighbour and similar operators enhance the image directly, maintaining the spatial relationship between pixcels. However, transforms such as Fourier or Walsh do not yield a recognisable image as a resulting matrix, as they transform to a different set of coordinate dimensions. Operations may be performed on such data in the transformed state, and the inverse transform may then be used to restore a modified form of the original image. The use of such transform pairs often provides a useful way of filtering or convolving images.

In this section a discussion is given of each of the more popular transforms, and some applications.

### 2.2.1.1 Smoothing

Smoothing operations are usually performed to reduce the effect of noise on an image, or to enhance the larger patterns in an image, and attenuate smaller patterns. Smoothing is also used to remove texture from an image (e.g. on grass) where there are a large number of adjacent light and dark points, that is where the image is "busy". In all of these cases data from adjacent pixcels over the entire image are merged, or compared and substituted, to make the average magnitude of rate of change between pixcel values lower.

**Nearest Neighbour**

The simplest forms of filter to implement are "Nearest neighbour" local filters. These filters simply replace the value recorded in each pixcel by the average value of a number of its (unmodified) neighbours, usually taken from a square grid "mask" of 9 or 25 pixcels, or a cross of adjacent pixcels. Examples are shown below of the mask matrix and weighting used in various types of filter.

| | | |
|---|---|---|
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |

(a) 9 Nearest Neighbour

| | | |
|---|---|---|
| 0 | 1/5 | 0 |
| 1/5 | 1/5 | 1/5 |
| 0 | 1/5 | 0 |

(b) 5 Nearest neighbour

| | | |
|---|---|---|
| 1/16 | 1/8 | 1/16 |
| 1/8 | 1/4 | 1/8 |
| 1/16 | 1/8 | 1/16 |

(c) 9 Nearest neighbour weighted

2.2 Near neighbour masks

The averaging given by the nearest neighbour masks shows a marked improvement in the smoothness of the matrix, The results of averaging using the matrices shown in figure 2.2 a) and b) are shown in figure 2.3 c) and d).

| | | | | | |
|---|---|---|---|---|---|
| 10 | 10 | 10 | 20 | 20 | 20 |
| 10 | 10 | 10 | 20 | 20 | 20 |
| 10 | 10 | 10 | 20 | 20 | 20 |
| 10 | 10 | 10 | 20 | 20 | 20 |
| 10 | 10 | 10 | 20 | 20 | 20 |
| 10 | 10 | 10 | 20 | 20 | 20 |

(a) Initial matrix

| | | | | | |
|---|---|---|---|---|---|
| 9 | 13 | 10 | 19 | 23 | 18 |
| 7 | 8 | 11 | 21 | 20 | 22 |
| 12 | 11 | 11 | 19 | 19 | 20 |
| 12 | 8 | 10 | 20 | 18 | 17 |
| 7 | 13 | 10 | 23 | 23 | 17 |
| 7 | 13 | 14 | 17 | 24 | 20 |

(b)Noise added

| | | | | | |
|---|---|---|---|---|---|
| 9 | 9 | 13 | 17 | 20 | 21 |
| 10 | 10 | 13 | 17 | 20 | 20 |
| 9 | 10 | 13 | 17 | 20 | 19 |
| 10 | 10 | 14 | 17 | 20 | 19 |
| 10 | 10 | 14 | 18 | 20 | 20 |
| 10 | 11 | 15 | 19 | 21 | 21 |

(c) Average 9 neighbours

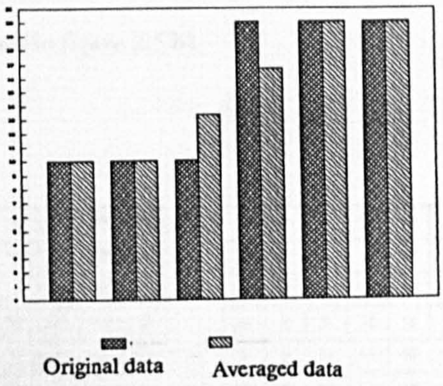
| | | | | | |
|---|---|---|---|---|---|
| 10 | 10 | 13 | 18 | 18 | 21 |
| 9 | 10 | 12 | 18 | 21 | 20 |
| 11 | 9 | 12 | 18 | 19 | 20 |
| 9 | 10 | 11 | 18 | 19 | 18 |
| 10 | 10 | 14 | 19 | 21 | 19 |
| 9 | 12 | 14 | 20 | 21 | 20 |

(d) Average 5 neighbours

2.3 Effects of near neighbour averaging

These filters are useful in reducing problems in boundary detection owing to noise present in individual pixcels. However, such filters tend to smooth edges as well as removing the incidental peaks that are produced by noise. This is frequently undesirable where accurate edge detection is needed.

Figure 2.4 illustrates this, the heavy shading represents a cross section of a matrix with a sharp step, the light shading represents the same cross section after nearest neighbour averaging.



2.4 The effect of averaging on a step function

It is possible to implement very fast versions of the square mask matrices if they have uniform weighting, by adding values from the pixcels that are included as the matrix moves across the image, and subtracting the (already calculated) value of pixcels that no longer belong to the matrix.

**Sigma filter**

A similar filter is the sigma filter, in which the value of a pixcel is replaced by the average of neighbours with values that are within a pre-specified tolerance range (t) around the original. The tolerance to be used can be estimated by taking the standard deviation of differences between adjacent pixcels as a guide. A range of values proportional to this is then taken as the tolerance range.

The results of application of the sigma filter to the data in figure 2.3 b) are shown in figure 2.5 a). The repeated application of the sigma filter to that in figure 2.5 a) is given in figure 2.5 b).

| | | | | | |
|---|---|---|---|---|---|
| 8 | 10 | 10 | 20 | 20 | 20 |
| 9 | 9 | 10 | 20 | 20 | 20 |
| 10 | 9 | 9 | 19 | 19 | 19 |
| 11 | 9 | 10 | 20 | 19 | 18 |
| 8 | 12 | 11 | 21 | 21 | 18 |
| 7 | 12 | 13 | 15 | 22 | 21 |

(a) Sigma filtered

| | | | | | |
|---|---|---|---|---|---|
| 9 | 9 | 9 | 19 | 19 | 19 |
| 9 | 9 | 9 | 19 | 19 | 19 |
| 9 | 9 | 9 | 19 | 19 | 19 |
| 9 | 9 | 9 | 19 | 19 | 19 |
| 9 | 9 | 10 | 19 | 19 | 19 |
| 9 | 10 | 11 | 12 | 19 | 19 |

(b) Sigma filter applied to (a)

2.5 Results of sigma filtering

**Median filter**

Median filtering goes one stage further, in that the value calculated for a pixcel is taken as the median value of nearest neighbours. This is the value of the pixcel that is nearest to (or lower than) the centre value of the set of nearest neighbours when sorted into numerically increasing order. This will tend to eliminate any great deviations from the norm in a mask region as they will never form the median value, and has the advantage that edges will be less blurred than by the averaging method. An example of the results of median filtering the data in figure 2.3 b) can be seen in figure 2.6.

| | | | | | |
|---|---|---|---|---|---|
| 7 | 9 | 11 | 19 | 20 | 20 |
| 9 | 11 | 11 | 19 | 20 | 20 |
| 7 | 11 | 11 | 19 | 20 | 19 |
| 11 | 11 | 11 | 19 | 19 | 18 |
| 7 | 10 | 13 | 17 | 20 | 18 |
| 7 | 10 | 13 | 17 | 20 | 20 |

2.6 Results of median filtering

The principal problem in using the median filter is that it takes a great deal longer to calculate than any of the methods described previously, since it requires $N \log_2 N$ comparisons for an N element mask matrix.

**Low-pass filter**

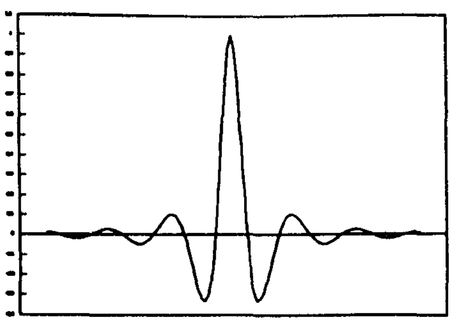
A conceptually direct method of smoothing an image is to use a low-pass filter. For the purpose of the following discussion a transformed image X' is defined:-

$$X' = F'(H * F(X))$$

where F( ) is the Fourier transform of an image, X is an image, F'( ) is the inverse Fourier transform and H is a weighting matrix which attenuates the higher frequency components.

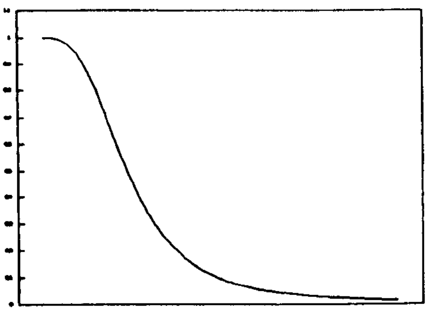
The 'Ideal' filter is one in which all components above a given cutoff frequency are completely removed. This is represented on a Fourier transform matrix for cutoff frequency $f_c$ by setting $H_{(i,\,j)}$ to zero for all $(i^2 + j^2) > (f_c)^2$,.

This filter will cut off all frequencies above that defined by $f_c$. Noise can be significantly reduced by this method, but it will also introduce "ringing". This effect is caused by the nature of the spatial characteristic of a filtered peak. The cross section of a transformed, filtered and inverse transformed impulse function is shown in figure 2.7. It can be seen that if there is a particularly bright region in the image then there will be rings appearing around that point in the filtered output. Similar results may be obtained from square filters where $H(i, j)$ is set to zero if both i and j are greater than $f_c$, or any filter around the origin with a sharp cutoff.



2.7 Effect of ringing produced by a filtered impulse

A better filter characteristic is the Butterworth filter (see figure 2.8). The advantage of this characteristic is that it does not show ringing on the transformed matrix after inverse Fourier transformation. This is attributed to the components of high frequency that are retained.



2.8 Butterworth filter characteristic

**Addition of multiple frames**

A further technique for removing noise is to average multiple frames of the same image. This is a very simple technique and will remove a great deal of stochastic noise. For a signal g(x, y) composed of noise free data f(x, y), and noise $\eta(x, y)$, where

$$g(x, y) = f(x, y) + \eta(x, y)$$

if the signal is averaged over M different examples, the standard deviation of $\bar{g}(x,y)$ from f(x, y) will be $\sigma_{\bar{g}(x,y)}$,

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{M}} \sigma_{\eta(x,y)}$$

where $\sigma_{\eta(x,y)}$ is the variance of the noise at each point.

It is not usual for identical images to be available for this averaging, however Keren, Peleg and Brada (see section 2.3.2) present an approach to registering frames from moving cameras, and this or similar methods may be used.

**References:-**

Rosenfeld[3], p.88
Gonzalez & Wintz[6], p161-173
Niblack[4], p.77

#### 2.2.1.2 Sharpening

Sharpening of detail in images is used in cases where blurring has caused exaggerated loss of fine detail. It has the drawback that it will tend to accentuate any high frequency noise in the image, however it is useful for edge detection algorithms, or where as much fine detail as possible is required.

**Gradient**

The most popular method of sharpening is to use a gradient transform. The gradient at any point is a vector representing the direction of greatest rate of change of intensity level, and the value is simplest to calculate at the points where four pixcels meet at one corner. The vector direction of the resultant matrix is largely unused in practical recognition schemes, however the magnitudes of the vectors gives a very straightforward edge detection system and are extensively used in those studies in which noise is not a major problem.

A more detailed discussion of gradient calculation algorithms is given in section 2.2.2.2 on the use of edge detectors.

**Laplacian**

The Laplacian transform is given by

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

which approximates in digital form to

$$L(i,j)=4\,a_{(i,j)}-(a_{(i-1,j)}+a_{(i+1,j)}+a_{(i,j-1)}+a_{(i,j+1)})$$

This corresponds to a mask centred on pixcel (i, j) of the form shown in figure 2.9 a), the integral of the whole being zero. It can be seen that this will tend to identify single peaks best, and will bring out edges to a lesser extent. Unvarying surfaces in an image will give a zero response. The mask in figure 2.9 b) is used as an alternative to that in figure 2.9 a) by some authors.

The Laplacian transform is not as good as the magnitude of the derivative for finding edges in an image because it is far more susceptible to point noise, and there appears to be little reference made to it for use in practical schemes.

| | | |
|---|---|---|
| 0 | -1 | 0 |
| -1 | 4 | -1 |
| 0 | -1 | 0 |

(a) 5 Cell
Laplacian Mask

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

(b) 9 Cell
Laplacian Mask

2.9 Laplacian filter masks

**High-Pass Filter**

The high-pass filter can be used in a manner similar to that described above for the ideal low-pass filter and the low-pass Butterworth filter. The differences are that the low frequency components are attenuated or removed rather than the high frequency components.

The Butterworth filter characteristic is described by

$$H(l,j)=\frac{1}{1+\left(\frac{f_c}{f_{(l,j)}}\right)^{2n}}$$

where $f_c$ is the frequency at which the input value is halved, and n determines the sharpness of the response cut-off.

**References:-**

De Preist and Wegman[5], p.94-100
Gonzalez and Wintz[6], p.81, 176-180

### 2.2.1.3 Fourier Transforms

The Fourier Transform of a signal is a continuous function giving the magnitude of every frequency component against frequency in the range minus infinity to plus infinity.

$$F(w, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x,y) e^{-j2\pi(wx+vy)} dx\, dy$$

The discrete form is given by

$$F(x, y) = \frac{1}{n} \sum_{p=0}^{n-1} \sum_{q=0}^{n-1} f(p,q)\; e^{-j2\pi(px+qy)/n} \quad 0 \le x,y \le n-1$$

In the discrete case such as with digitised information the output consists of a series of values representing the magnitude of response for individual frequencies.

In the two dimensional discrete Fourier transform the response frequency for a cell g(x, y) is proportional to the distance of the cell from the origin ($\sqrt{x^2+y^2}$). The direction of waves indicated by g(x, y) is the same as the direction of the cell (x, y) from the origin (0, 0).

A major advantage of the Fourier transform over other algorithms is that the properties and method of transformation are extensively documented, and the Fast Fourier Transform algorithm requires only $N \log_2 N$ operations to calculate, rather than $N^2$.

As has already been seen in the sections on smoothing and sharpening above, the Fourier transform may be used to provide a simple filter to enhance an image. The filter used may be designed either by empirical means using graphical display of the Fourier domain and a "good guess". Alternatively an optimal filter may be constructed if the noise is independent of frequency, by obtaining a good image using other means (eg. averaging or operator directed enhancement), and taking the Fourier transform of this as the filter, thus enhancing the image and reducing the noise.

Another use of Fourier Transforms is for correlating two images. The convolution theorem gives us a means of locating the best fit in the spatial plane for a pattern. The method is to Fourier transform an image, and Fourier transform the search pattern, multiply the two resulting matrices, and then to inverse Fourier transform the result. This gives a peak value at the point of greatest registration of the search pattern with the image. Therefore the Fourier transform may be used as a means of efficiently finding the best correlation between two images.

**References:-**

Niblack[4], p.95
Gonzalez and Wintz[6], p.100, 307

#### 2.2.1.4 Walsh / Hadamard transforms

The Walsh transform is in many respects similar to the Fourier transform. The principal difference is that whilst the underlying correlating function in the Fourier transform is a sine wave, the correlating function in the Walsh transform is a square wave, see figure 2.10 a). This has the immediate benefit that no floating point multiplications are needed, thus the time required to find the Walsh transform will be much less than that for the Fourier Transform.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| +1 | +1 | +1 | +1 | +1 | +1 | +1 | +1 |
| +1 | +1 | +1 | +1 | -1 | -1 | -1 | -1 |
| +1 | +1 | -1 | -1 | +1 | +1 | -1 | -1 |
| +1 | +1 | -1 | -1 | -1 | -1 | +1 | +1 |
| +1 | -1 | +1 | -1 | +1 | -1 | +1 | -1 |
| +1 | -1 | +1 | -1 | -1 | +1 | -1 | +1 |
| +1 | -1 | -1 | +1 | +1 | -1 | -1 | +1 |
| +1 | -1 | -1 | +1 | -1 | +1 | +1 | -1 |

(a) Walsh Kernel, N = 8

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| +1 | +1 | +1 | +1 | +1 | +1 | +1 | +1 |
| +1 | -1 | +1 | -1 | +1 | -1 | +1 | -1 |
| +1 | +1 | -1 | -1 | +1 | +1 | -1 | -1 |
| +1 | -1 | -1 | +1 | +1 | -1 | -1 | +1 |
| +1 | +1 | +1 | +1 | -1 | -1 | -1 | -1 |
| +1 | -1 | +1 | -1 | -1 | +1 | -1 | +1 |
| +1 | +1 | -1 | -1 | -1 | -1 | +1 | +1 |
| +1 | -1 | -1 | +1 | -1 | +1 | +1 | -1 |

(b) Hadamard Kernel, N = 8

2.10 Walsh and Hadamard Kernels

It is not easy to find a use for the Walsh transform, since it is not shift invariant, thus it cannot be successfully used for correlation of two items, nor can it be used as a pseudo frequency-domain filter.

The Hadamard filter is very similar to the Walsh transform, but the pattern of +1 and -1 is shuffled with respect to the Walsh transform, this is illustrated in figure 2.10 b). The comments above about the Walsh transform apply equally to this transform.

**References:-**

Gonzalez and Wintz[6], p.111-115

#### 2.2.1.5 Hough Transforms

The Hough transform is a means of determining the best straight line that passes through a set of points. The equation of a straight line

$$y = mx + c$$

may be rearranged to give

$$c = y - mx$$

Thus for any point (x, y), the value of m may be varied over its permitted range, giving a series of values for c. These values of m and c are accumulated in a 2-dimensional array. The process is repeated for all points in the image containing a potential edge, and the accumulator cells in the (m, c) matrix with the highest values are those that are most likely to represent straight lines.

For practical reasons it is more convenient to represent straight lines by the equation

$$r = x \cos \omega + y \sin \omega$$

since both $\omega$ and r are constrained to be within known limits, and may conveniently be stepped linearly throughout these limits.

The transform may be extended to include any shape that can be expressed parametrically in the form

$$r = g(x, y, p_1, p_2, \ldots, p_n)$$

in which case an (n+1) dimensional accumulator matrix is required. However, this is not usually practical for cases where n is greater than 2 since it requires $n^2$ calculations for each point.

An example of this is for points lying on a circle, described by

$$r^2 = (x + p_1)^2 + (y + p_2)^2$$

which would give a 3 dimensional matrix containing the most likely x position, y position and radius for circles in an image.

**References:-**

Gonzalez and Wintz[6], p.130
Ballard and Brown[2]

### 2.2.2 Segmentation

The process of segmentation is that of dividing an image into segments. A segment is a region in which it is possible to go from any cell within the region to any other cell, at all stages remaining inside the region. This is a fully connected region. It is permissible for a region to wholly contain a different region. The process of segmentation does not include the association of any meaning with segments, though some a-priori knowledge of the meaning of the segments is usually useful in developing the segmentation algorithm.

#### 2.2.2.1 Thresholding

The process of thresholding involves the selection of a grey level value or values that separates one segment from another. The most common method of selecting the value in research is by inspection, and in automated systems a great deal of effort is spent in ensuring that the threshold value is as easy to select as possible. However, certain applications necessitate the selection of a value by automatic means.

**Fixed value**

A fixed value threshold $t_0$ is defined

$$F(x,y) = \begin{cases} 1 \text{ iff } f(x,y) > t_0 \\ 0 \text{ iff } f(x,y) \leq t_0 \end{cases}$$

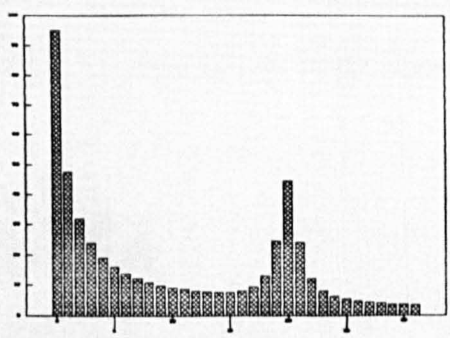
It is unusual for a fixed value to be appropriate in any except the most closely controlled conditions, since any variation in lighting or the reflectance of objects within the scene is liable to cause incorrect segmentation. The fixed value threshold is therefore appropriate in cases where an object is illuminated from behind, or for a particularly bright object on a dark background.

**Percentile**

The percentile threshold method is suitable for cases where the ratio of segment to non-segment values is known in advance, e.g. because the area of the object is known, but not its location or orientation. The method is to chose the threshold such that the overall ratio of segment to non-segment cells is maintained. This technique will not preserve edges perfectly if noise is present in the image, because no account is taken of the gradient of edges about the threshold value, thus for example noise of 4 units on an average gradient of 2 units per pixcel will be likely to give many points that are separated from the main region. This can, however, lead to a "correct" segmentation suitable for further processing (see section 2.2.2.4 Relaxation).

**Mode**

The mode method is based upon the assumption that any region to be segmented is the only region in the image that has a given set of grey levels. In the ideal case the histogram of "number of pixcels of brightness g" versus "brightness g" will be as shown in figure 2.11. The threshold is chosen to be between the peaks, thus only a relatively small number of pixcels will be misinterpreted.



**2.11 Histogram showing grey level modes**

Problems with this method occur when the valley between the peaks is not sufficiently well defined, and when there are several peaks in the histogram. Several researchers have proposed methods to overcome these problems, by sharpening the peaks in the histogram, by ignoring pixcels close to steep gradients, and by moving values on the histogram to adjacent locations, creating better definition.

**Variable thresholds**

Using a threshold that varies over the image can overcome several problems. A method is described in chapter 4 that uses a-priori knowledge about the image to create a threshold that varies linearly across the image. Other techniques are to use local averaging (eg. over 1/9 of the image area around the pixcel) to calculate a threshold for each pixcel, or to compare the pixcel with the median of 9 or 25 nearest neighbours. The danger with all of these techniques is that noise imposed on a relatively flat region will cause a great deal of busyness that would not be selected if the threshold were chosen by an operator, because the threshold will tend to the value of the flat area.

**Bayesian Maximum Likelihood**

The Bayesian Maximum Likelihood method is similar to the percentile method. The probability that any pixcel belongs to a given region is calculated for each pixcel, and does not necessarily have to remain uniform across the whole image for any given pixcel value. Having calculated the probability that a pixcel belongs to any of the possible region types (there may be more than two), the pixcel is then assigned to the region for which the probability is highest. If the probability is dependent upon adjacent pixcels several iterations may be required to determine a complete result. Estimates of the probability functions are required before the classification is carried out, and histograms such as are used in the Mode method of classification are frequently used in this case. (see also section 2.2.2.1 Multiple Thresholds, below).

Typically the probability will be based on a combination of grey level, percentile area, the classification of near neighbours and wherever possible a-priori knowledge of the image. It would also be reasonable to use a-posteriori knowledge in an iterative recognition scheme.

**Multiple thresholds**

To overcome the problems of variable and modal thresholds the use of multiple thresholding has been developed. Two threshold values are chosen, above the upper threshold a pixcel is definitely assigned to one region, and below the lower value the pixcel is definitely in the other region. These thresholds may be chosen using a-priori knowledge in the same way as single thresholds.

All pixcels that fall between the two thresholds must be defined in terms of probability of belonging to one region or the other by inspection of their neighbours, or by probability weighted by neighbours. For example it would be likely that a pixcel entirely surrounded by a single region also belongs to that region, though a pixcel surrounded by four pixcels of one region and four of another must be defined by other means, such as local average, gradient or a-priori knowledge. It is usual that several iterations of the algorithm are required, firstly to fully define any large areas, and secondly to avoid problems that occur due to the order in which pixcels are evaluated.

**References:-**

Niblack[4], p.113
De Preist and Wegman[5], p.128, 132
Gonzalez and Wintz[6], p.354
Pavlidis[1], p.66

### 2.2.2.2 Edge detection

A number of techniques approach the problem of object location by attempting to locate the edges of an object pattern, and then combine the edges that are found into a complete pattern.

For our purposes an edge is defined as a region where the rate of change of value with distance across the input matrix is greater than a prespecified threshold. Usually the measured value is the brightness at a point, but texture, colour or other attributes that may be transformed to a matrix of point values, could equally be used.

Advantages of edge detection are that it can be less sensitive to global variations across an image, since it is specifically concerned with local information, and that the object located does not need to be a complete closed contour description. This latter may be considered a disadvantage by many researchers, however, in section 2.2.2.2 methods are demonstrated which can link edges into related sets, closing contours where required.

**Magnitude of gradient**

The gradient operator is the most popular, and obvious candidate for edge evaluation. This arises directly from the definition of an edge being a binary measure of rate of change of value.

Several methods for measuring the gradient at a point have been proposed. Taking the difference between a pixcel and its neighbour at a point will give a simple method, however this is susceptible to noise, and usually a mask of 3 x 3 or more pixcels is employed.

The Sobel operator is widely used for gradient evaluation, see figure 2.12. The response for a flat region will be zero, and the response for a gradient of 1 grey level per pixcel will be 8 units. It should be noted that using a 3 x 3 mask will tend to spread the edge, and that even a simple uniform step will produce a response of 2 pixcels width. Using a 2 x 3 operator overcomes this problem, but gives a problem of combining values of gradient for the x and y directions that cannot be evaluated for a single point. Using 2 x 4 operators to determine the gradient over a larger area for a single point at the corner of four pixcels is a reasonable alternative.

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

(a) Y gradient

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

(b) X gradient

2.12 Sobel masks

The magnitude (M) of the gradient is strictly $M = \sqrt{R_x^2 + R_y^2}$, however, it is common practice to simply sum the components to give $M = |R_x| + |R_y|$ to determine the approximate gradient at a point. This is acceptable since it is not the precise magnitude of an edge, but the presence of the edge that is usually required. In the worst case the measured gradient will be multiplied by a factor of $\sqrt{2}$ for edges that run at 45° to the detector grid when compared with the same edge running parallel to the grid, but most edge following algorithms will locate the peak edge even if it is not uniform in magnitude along its length.

**Direction**

The direction of the edge is given by $\tan^{-1}(R_x/R_y)$. The precision of direction that this affords is not often useful, and most edge following algorithms will cast around a known edge pixcel for the largest edge on the current heading and not use the measured direction of the edge at all. The direction is however used by some algorithms and there is no suitable short cut for calculation.

**Line detector**

The line detector operator given in figure 2.13 corresponds to the second derivative of the original matrix multiplied by -1. This operator is very straightforward to implement, but is only useful for particularly noise free images, since it tends to enhance noise.

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

(a) Horizontal

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

(b) Vertical

2.13 Line detector masks

**Laplacian**

The Laplacian operator (also described in section ) is a second derivative point detector. Two masks are proposed by various authors as shown in figure 2.9. The same problems as with the line detector occur with this operator, and there are few documented uses of this operator except as a means of verifying the presence of an edge.

**Edge Following**

In many instances it is possible to find several partial edges but with sections of the edge missing, or to have determined a start point to an edge which should be continued. In these cases edge following or filling algorithms are used, or the Hough transform (see section 2.2.1.5) is applied.

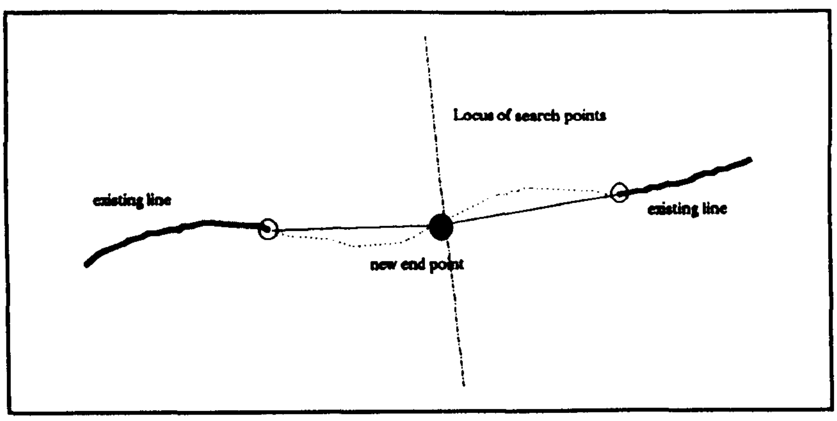
Edge following algorithms use the a-priori knowledge that most edges tend to extend along in the same direction with only small changes in direction when looked at over a small length, thus a weighting may be given at the end of known lines to enhance any edges that follow or deviate only slightly from the known edge. Thus edges that would not normally be sufficient to reach a threshold value on the first derivative of an image may be enhanced to do so.

It is also known that the measured direction of the gradient will not vary rapidly, and may indicate an adjacent pixel in a direction perpendicular to the direction of the gradient vector that should have its value increased to become the next pixcel that should be tested as an edge. If both edge magnitude and direction together agree that a pixcel should be included, then this is further evidence for the pixcel to be included.

An enhanced form of edge following comes from the use of a technique of dynamic programming presented by Ballard & Brown [1] p.137. It would not be practical to calculate and record the likelihood of all edges in an image that start at a given point and extend over a large distance, since the number of paths would increase exponentially with the distance from the start point. However it is possible to record the most probable path to any intermediate point. The most probable path to a point one pixcel further distant may then be found using the probabilities of the paths to adjacent cells and the edge probability function for the new cell.

**Edge finding between points**

It is frequently found that two edge sections are co-linear, or nearly so, but that there is a missing edge section between them. In this case the problem of edge location is greatly simplified, the method is to look for the largest gradient vector with a direction similar to that of the known end points, that lies on the locus of points equidistant from the end points (see figure 2.14).




2.14 Edge finding between known end points

It is supposed that in most cases this pixcel will not be far from the line that connects the end points, and thus will be reasonably easy to locate. This point then forms a known end point, and by determining the direction of the largest slope a further approximation to the required edge may be found. The process of locating the edge between the known end points and the new point is then repeated until the entire edge section has been found.

**Use of a-priori knowledge**

Use of a-priori knowledge has also been shown to be advantageous in the case of edge following, however there may also be global knowledge that can be used. For example in medical imaging of blood cells for blood counts, the size of and shape of the patterns is known, thus any edge detection may be enhanced by convolution with the known shape.

**References:-**

Niblack[4], p.117
Gonzalez and Wintz[6], p.256, 275-289, 333-334
Ballard and Brown[2], p.121, 131-143
Pavlidis[1], p.67, 142

### 2.2.2.3 Region Descriptions

Once a region has been found it must be represented for further processing. The simplest representation is a binary 1 or 0 in each cell of a matrix, giving the presence or absence of the region. This representation has many problems, however, as it is not independent of position, size or orientation, and is thus not easy to match with other similarly shaped patterns. The storage requirements are also particularly great since it may be that the original image matrix is much larger than the segmented pattern, and thus much redundant data will be stored.

Two approaches to pattern representation are frequently used. The first is a description of the boundary, the second is a representation of a collection of pixcels within the boundary. Several schemes are described below which use one or other of these methods. In general the schemes for description in terms of boundary are less sensitive to changes in orientation than region descriptions, however, the regions are more suitable for subsequent syntactic analysis, since they retain the relationships between any separate identifiable parts directly.

**Chain codes**

Chain codes represent the edge of a pattern in an image by first converting the contour into a series of short line segments which may lie along a limited number of directions. Four, six or eight directions are usually allowed, and each direction is denoted by a number. The pattern then may be described by a chain of codes representing the vectors used to approximate the edge. The representation may be as close as desired, but in the coding where only four directions are allowed, lines at $45^{\circ}$ to the digitising grid are replaced by a set of codes giving a stepped effect, thereby mis-recording the line length and thereby losing rotational invariance. The coding which allows eight directions is better in this respect, but the vector length is not the same for all of the vectors. In both of these codings it is possible to nearly reconstruct the original figure, however, correlating the chain code representing an object with the chain code representing the same object rotated through less than $90^{\circ}$ will fail to give a good correlation.

**Bearings and psi curves**

An extension to the strategy of chain codes is to represent a curve (not necessarily closed) by a series of fixed length vectors. The angle between the line joining two adjacent points on the pattern boundary and a fixed direction is recorded as a data value for each pair of points along the curve. The accuracy of this description is limited only to the number of pairs of points used and the accuracy of representation of the measured directions.

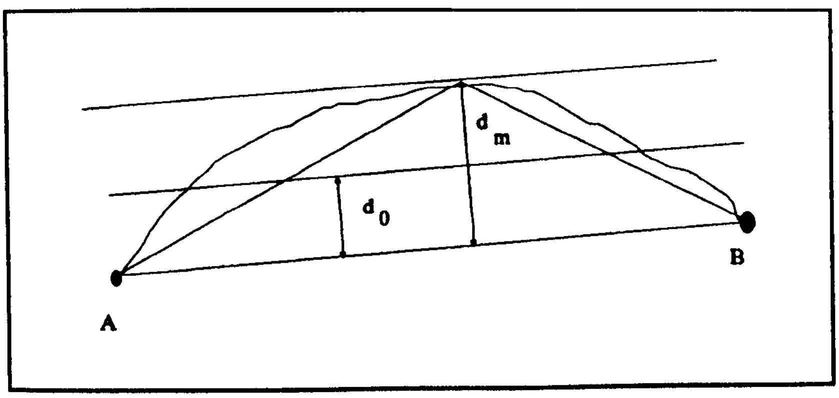
However, this approximation is prone to noise from incidental islands and voids along the periphery of the object, thus smoothing should be performed before and/or after the bearings have been found. It is also possible that due to approximations in locating points along the curve the resultant description will not produce a closed contour where one would be expected, however this does not detract from the accuracy of representation unless a closed contour must be produced.

Psi curves are derived from these data values by calculating the first derivative of the set of values, and collecting together the regions where this is constant (or within a specific range of values). These sets then form regions of constant curvature including straight lines and any region may be represented by its length and average curvature alone. Thus a compact description of an object may be found to any required degree of accuracy.

**Polygonal Approximations**

Any curve may be represented to a given degree of accuracy by a series of straight lines. The required accuracy may be attained by ensuring that the curve that must be approximated is never more than a pre-specified distance ($d_0$) away from the approximation, and the approximation is divided into two parts if the threshold is exceeded (see figure 2.15).

Difficulties with this approach come from noise on the curve which may create false points when the curve comes close to exceeding the threshold. A solution to this would be to use a smaller threshold than ultimately needed, then group the vectors so found into sets that are as evenly sized as possible, whilst not violating the required threshold.




2.15 Polygonal approximation to a curve

**Polynomial approximations**

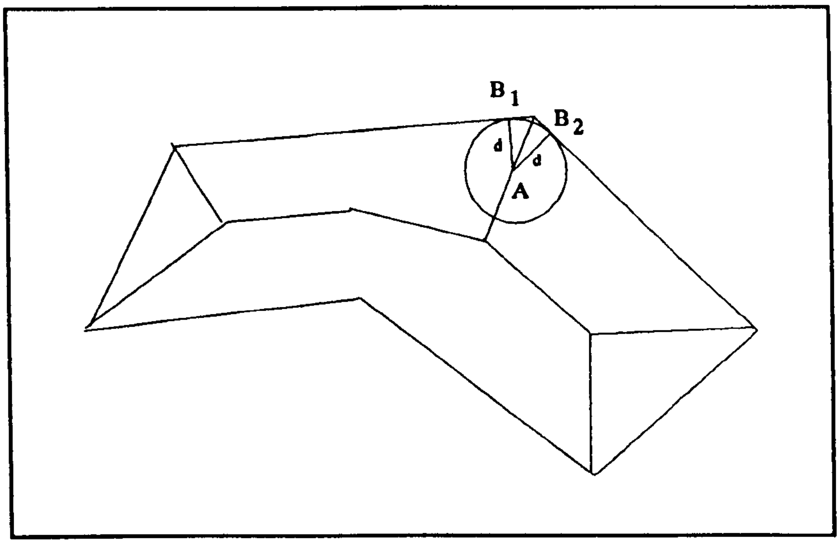
The use of straight lines as approximations to curves can be extended to using polynomials as approximations. Methods proposed are to used splines to generate smooth approximations to the entire curve, or to use conic sections to model parts of the original. An advantage is that the resulting curve "looks good" as a smoothed example of a noisy original, however, the curves are difficult to generate mathematically, requiring many iterations of a long calculation. Also there is no consensus of the best function for calculating the splines, and authors of texts covering many fields of pattern recognition who have not implemented this type solution have tended to avoid the detailed mathematics describing the calculation and behaviour of these functions.

**Fourier Descriptors**

Fourier descriptors are the Fourier transform components of a continuously measured parameter around the boundary of an object pattern. There are several useful parameters that may be used, for example the vector between a fixed point and points at intervals along the boundary. The author has used the rate of change of direction of the boundary itself, and this is described in chapter 5.

**Skeletons**

The skeleton of a region is produced either by determining points within a region where the minimum distance to the region boundary touches the boundary at more than one point, or by successively removing pixcels from the boundary until there is only one line of pixcels left.




2.16 Skeleton of a polygonal region

Figure 2.16 shows a polygon with its skeleton drawn in. The distance A-$B_1$ is equal to the distance A-$B_2$, thus point A is on the skeleton of the polygon.

Using the method of checking each point is conceptually the easier method, however it can be very expensive in terms of computer time, since it involves finding the shortest distance to the boundary and then checking in a circle to verify that it touches but does not cross in at least one other place. This must be done for every point within the polygon. Alternatively the distance from every boundary pixcel to the sample point may be calculated, and the two smallest tested for equality. In either case a large number of floating point calculations must be performed. This algorithm may be speeded up by determining a single point that forms part of the skeleton, and then "growing" the skeleton by testing all of the adjacent points. This procedure is then repeated for each skeleton point found until the skeleton limbs have all reached the boundary.

The method of determining the skeleton by thinning the region is less expensive in terms of computer time, but has the disadvantage that the limbs of the skeleton must be artificially preserved from being truncated, and this requires identification of the end points of limbs prior to the thinning.

A solution to this problem is to apply the test for a pair of shortest distances to the boundary for each pixcel that is a short distance inside the boundary. Points which are found by this test are then chosen as the end points to the skeleton limbs. This method will only find limbs where the radius of curvature of the boundary is less than the shortest distance of the tested point, thus it may be that not all limbs will be found.

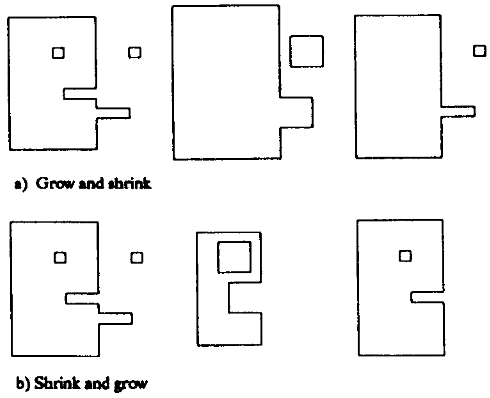
A significant drawback of skeletons as a method of region representation is that it is not always possible to reconstruct the original image. As an extreme example, the skeleton of a circle is a single point, and there is no information held about the radius of the circle.

#### 2.2.2.4 Relaxation

An improvement in the basic segmentation may be achieved by reassigning pixcels to segments after the initial segmentation. Usually the reason for this is to reduce the effects of noise in the image. These will be shown by regions that are isolated from the main body of a pattern, and by small holes within the pattern. It may also be found that a more efficient classification of regions can be made using the a-posteriori knowledge of the classification of other pixcels, rather than attempting a classification of all pixcels in a single pass.

**Shrink and grow**

The shrink and grow transformation is used to remove small features from a pattern. The shrink is defined by setting all eight neighbours of a background point to a background value. The grow sets all points adjacent to an object pattern point to be included in the pattern. It can readily be seen that any object of less than 2 pixcels in width is lost, as are any convex features on the pattern boundary.




2.17 Shrink and grow operations

**Grow and shrink**

The same operations as shrink and grow applied in reverse order do not have identical effect. In this order any small point on the border of a pattern is retained, but any small hole within the object is lost.

## 2.3 Specific Research Studies

### 2.3.1 A Modified Scheme for Segmenting the Noisy Images.

*B. Chanda, B.B.Chaudhuri, D. Dutta Majumder* [10].

#### 2.3.1.1 Abstract - from the paper

*An image segmentation scheme based on gray level thresholding is presented. To reduce errors in misclassification, grey level histograms are sharpened before thresholding using a gray level transformation function that also leads*

*to an expression for computing the expected threshold. Three new thresholding methods are proposed that reduce the noise, smooth region boundaries, and preserve connectedness among different parts of objects, and are not expensive.*

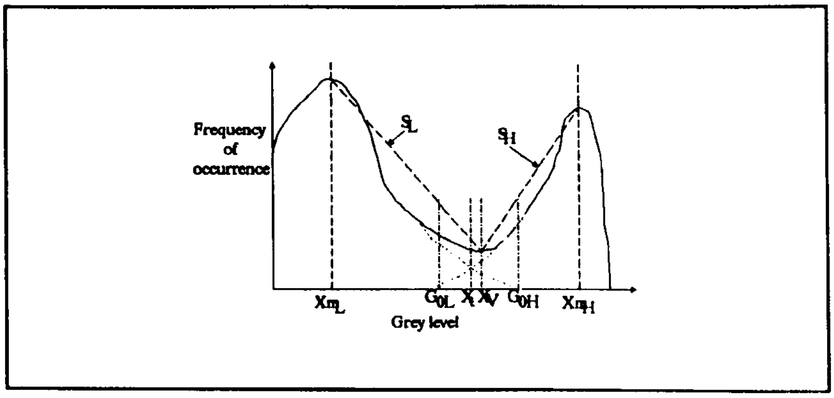
**2.3.1.2 Method**

The method described is suitable for image g(j, k) that contains a pattern representing an object ($S_i$), where the pattern and background are at significantly different grey levels. The segmentation aims to classify all pixcels in the image into two regions. The approach is a modified form of the Mode method of segmentation.

The histogram of "frequency of occurrence of grey level" versus grey level is first produced ($h_g(x_i)$). This is then smoothed by taking the average of values for points of $\pm q$ grey level values each side of every point to produce $\bar{h}$.

$$\bar{h}_g(x_i) = \frac{1}{2q+1} \sum_{l=-q}^{+q} h_g(x_{(i+l)})$$

This averaging reduces the probability of local maxima interfering with the location of the mode maxima. It does, however, tend to reduce the ratio of $\max(g(j,k)) / \min(g(j,k))$ thereby making the problem of finding the correct minimum point between the modes slightly harder.




2.18 Histogram of bimodal image data

End points are then chosen for the low and high end of the histogram ($H_L$ and $H_H$) such that all points outside this region are defined as zero. The remaining region is divided into K parts of size s, where

$$s = \frac{H_H - H_L}{K}$$

and K is a constant chosen a-priori, and local maxima are found within each part (g), such that

$$\bar{h}_g(x_l) = \max_{l=-s}^{+s} (\bar{h}_g(x_{(l+l)}))$$

The location of the minimum point between the lowest and the highest grey level maxima is next found ($X_V$), and the largest maxima of grey levels less than this value ($Xm_L$) and greater than this value ($Xm_H$) are located. Figure 2.19 illustrates the values described.

A measure of the bimodality of the histogram is then determined by measuring the slope of each that connects the peaks with the trough, marked $S_L$ and $S_H$ on figure 2,19). If both lines rise to peaks then the histogram is said to be bimodal, if the gradient of the lines are both greater than a pre-specified minimum then the histogram is strongly bimodal. The pre-specified minimum gradient is dependent upon the sampling density, and the number of grey level available, however guidelines are not given in the published paper to determine appropriate values.

In the case where one gradient is zero or negative the histogram has a shoulder and will not be sufficiently bimodal for determining a threshold.

The next stage of the operation is to sharpen the histogram to improve the estimate of the threshold value ($x_t$ on figure 2.19). This is performed by nearest neighbour averaging, or by median filtering for all pixcels in g(j, k) with grey levels in the range $G_{OL}$ to $G_{OH}$, where $G_{OL}$ is the theoretical lowest value of the high mode (object pattern) pixcels, and $G_{OH}$ is the highest value of the low mode (background) pixcels. If the value of any specific pixcel is increased, then it is included in the high mode (object pattern) by adding a small amount to the grey level to reflect this change. Similarly if the value of a pixcel is reduced, a small amount is subtracted to draw it into the lower mode (background) region. For those cases where the value of the pixcel does not change with filtering, the value of the pixcel is compared with the grey level ($X_V$), and modified in the same way as for all other pixcels, that is it is reduced if lower than or equal to $X_V$, and increased if higher than $X_V$. It is probable that several iterations will be required to force all pixcels into one mode region or the other.

An alternative method of improving the histogram is also presented. This involves counting twice all pixcels that lie within an object or within the background, but that do not lie on the border. The border pixcels are defined as those between $G_{OL}$ and $G_{OH}$. This in turn depends upon having determined values for these thresholds, thus the histogram improvement may only be performed as part of an iterative scheme.

The pattern is finally extracted from the image by segmenting the image about the threshold value ($X_v$) that has been determined, using the values for the sharpened matrix which force pixcels greater than $X_t$ to be greater than $X_v$, and pixcels less than $X_t$ to be less than $X_v$. It is suggested by the authors of the paper that a nearest neighbour average or a median transform should be performed on the image before thresholding, to reduce the busyness of the pattern.

Three methods are also shown for performing the thresholding, in particular avoiding isolating parts of an object.

The first approach is to assign pixcels to object pattern or background by inspection of the pixcel and its eight nearest neighbours. If four or more of the nine pixcels are above the threshold, then it is ascribed to the pattern, otherwise it is ascribed to the background.

The second approach is an extension of the first, except for the case where exactly four of the eight near neighbours are below or equal to the threshold value, and the centre pixcel of the 3 x 3 set is below the threshold value. In this instance particular care is taken to avoid the problem parts of the object pattern becoming separated from the main body. This is done by setting the centre pixcel to be part of the pattern region if there are two separate regions of pattern in the set of nine pixcels, and the shortest distance between then passes over the centre pixcel.

The third approach is similar to the second approach, except that the centre pixcel of a set of nine is always ascribed to the object if there are two or more separate parts of object in the scene. This method will connect parts of the object that the second method will not, but has the problem that it will also tend to build isolated spikes of noise into parts of an object where they should belong to the background.

#### 2.3.1.3 Problems

The method supposes that there is a single 'correct' threshold value. From the author's own work it has been found that where the single value threshold is used, any one of a range of values will produce a suitable division of object pattern and background. Using the mode method to determine this threshold value gives a simple and automatic means, however, any further processing to change this threshold by a small number of grey levels is unlikely to lead to a better segmentation.

An associated problem is the supposition that a single threshold is suitable for the entire image. It is more frequently found that due to variations in lighting or the image capture hardware the thresholding that achieves a uniform segmentation requires that the threshold value varies over the image. The author has used a linear variation , but this has been found to be inadequate for many of the images studied, and further thought must be given to the utility of this approach.

The use of a local filter to connect otherwise disconnected regions of an object is a useful technique from the experimental results presented in the paper, however, the third method presented suffers from a tendency to collect isolated points of noise near the object edge and extend the edge to include the noise. Similarly the second method frequently fails to connect regions that should be connected.

It is the opinion of the author that if it is not known a-priori whether two segments should be joined, and how they should be joined, then the best approach is to use a-posteriori information to join them together, if required, and that this information will only be available when a potential recognition has been made. For this reason it is not likely to be useful to force a segmentation to link object patterns and thus the second method of segmentation, which only makes use of a-priori information about local regions is a better approach that the third method.

## 2.3.2 Image sequence enhancements using sub-pixcel displacements.

*D.Keren, S.Peleg, R.Brada,* [11]

### 2.3.2.1 Abstract - from the paper

*Given a sequence of images taken from a moving camera, they are registered with sub-pixcel accuracy in respect to translation and rotation. The sub-pixcel registration enables image enhancement in respect to improved resolution and noise cleaning. Both the registration and the enhancement procedures are described. The methods are particularly useful for image sequences taken from an aircraft or satellite where images in a sequence differ mostly by translation and rotation. In these cases the process results in images that are stable, clean and sharp.*

### 2.3.2.2 Method

The approach taken by the author of the paper is to improve a set overlapping images by averaging (see also section 2.2.1.1). The method is first to accurately register images to a reference position, and then to average overlapping pixcels to produce a set of pixcels with reduced noise. The resulting image is then high-pass filtered to reduce the gaussian noise introduced by small registration errors, which tends to blur edges in the resulting image. A final step is to attempt to reduce the errors resulting from the image capture process by minimising the error with respect to a model of this process.

**Sub pixcel registration**

The registration process is performed by starting from the equation describing translation and rotation of an image.

$$g(x, y) = f\,(x \cos\theta - y \sin\theta + a, x \sin\theta + y \cos\theta + b)$$

This is expanded as a Taylor series, which gives a set of linear equations to solve for a, b and $\theta$. Non linear terms are ignored, thus this method will only work for small rotations and translations. The technique used to overcome this is to construct a low resolution image with a much smaller number of pixcels than the full image by averaging over a rectangular set of image pixcels to obtain the smaller image. This image is then low-pass filtered to improve the ability of the algorithm to register over a distance. The registration is then performed to find the minimum error value, and resulting rotation and translation distances are found. By multiplying the translation parameters produced by the large to small image scale factor, a true error distance may be found. The process is then repeated several times for a corrected image using a finer resolution each time until the full resolution image has been corrected.

**Averaging**

Once several images have been brought into a common registration an average of the images is taken. This is done by determining the centre position of each pixcel on the output matrix, and taking the value that corresponds most nearly in each of the corrected input images for the averaging process.

**Improvement**

The resulting image is slightly blurred, owing to errors in position correction and averaging processes. To overcome this the image is high-pass filtered giving a significant improvement.

A second part to the improvement is to model the characteristics of the image collection apparatus. A series of corrected and averaged images are transformed using the model, and compared with the input images. The output images are then altered by 1 grey level in one pixcel and the effect on the transformed image is found. If the overall error is decreased then a source of systematic error has been found and can be corrected for. This process is repeated for each pixcel in the output matrix.

**Discussion**

Stage one of the algorithm works reasonably well and the author of the paper states that images can be lined up to within 0.03 pixcels and $0.03^{o}$. The overall rotation must be small, however, usually less than $6^{o}$.

The averaging and high-pass filtering is less than optimal. It is clear from the approach used that regardless of the precision of registration the averaging assumes that all images are composed of a set of uniformly light pixcels, rather than a digitised representation of a continuous function. A better approach would be to perform the averaging in the Fourier domain, this would have the twin advantages that the averaging would be done by a method that correctly allows for the image frame, and any high-pass filtering that is required can easily be performed by multiplying by a suitable weighting function in the Fourier domain.

## 2.3.3 Edge Detection by Partitioning

*Jong-Sen Lee* [12]

### 2.3.3.1 Abstract - from the paper

*The objective of an edge operator is to detect the presence and location of grey level changes in an image. Various edge detection algorithms have been developed in recent years. In this paper a new edge detector is proposed based on the idea of separating pixels in a local window into two sets of similar intensities. Then the difference in intensity averages of these two sets is the edge magnitude. The technique of separating pixels in a local window into two sets is motivated by the Sigma filter, which is an image noise smoothing technique based on averaging only pixels within two standard deviations from the intensity of the centre pixel. The characteristics of this edge operator are its flexibility in setting magnitude of edges to be detected and also its consistency in detecting edge strength independent of edge orientations. Furthermore it can be easily generalised for surface detection in three dimensional images. This algorithm is also computationally efficient since only simple fixed point operations are involved.*

The sigma filter is described in section 2.2.1.1

### 2.3.3.2 Method

The approach described in this paper is an extension of Sigma averaging to determine the gradient. The method used is to assign a threshold ($t_h$) at $\Delta$ units above the value of the centre pixcel ($x_0$) of a 3 x 3 matrix, and a threshold ($t_l$) $\Delta$ units below this centre value. This gives 3 sets of pixcels within the matrix, those greater than $t_h$ (set A), those less than or equal to $t_h$ and greater that $t_l$ (set B), and those less than or equal to $t_l$ (set C). If the number of pixcels in set A is smaller than the number of pixcels in set C, then set A is merged with set B. If set A is larger than set C then set C is merged with set B. In each case the two resulting sets are averaged and the gradient at $x_0$ is the magnitude of the difference of the two averages.

A number of exceptions and modifications are made to the rules so far stated:-

*i)* If the number of pixcels in set A added to the number of pixcels in set C is less than 2, then the value of $\Delta$ is divided by a factor (usually 2), and the set contents re-calculated. This step ensures that the value of $\Delta$ is of sufficient resolution to fit the approximate magnitude of the gradient when the mask contents are relatively flat. This step may be repeated a number of times until $\Delta$ reaches a pre-set minimum value.

*ii)* If the number of pixcels in set A added to the number of pixcels in set C is 8, then $\Delta$ is increased by a factor (usually 2). This ensures that $\Delta$ is appropriate to any large change in grey level that may be found in an image. This step may also be repeated up to a pre-set maximum value.

*iii)* If set B contains only one pixcel, and set A or set C is empty, $x_0$ is an isolated peak in the image, and the gradient is given as 0.

*iv)* If set A and set C contain the same number of pixcels, then either set may validly be merged with set B, there will be no difference either way.

The approach outlined above is computationally efficient, and compares favourably with the Sobel operator for gradient detection. An advantage is that by setting the minimum $\Delta$, edges below a preset value may be ignored. A disadvantage is that the direction of the gradient is not supplied by this algorithm, however, in many instances this is not required.

The algorithm may be generalised to n x n matrices, but care must be taken when dealing with the connectivity of sets of points. The author of the paper proposes that only the largest connected region within each of the sets A, B and C are used, and that any other data points should be ignored.

## 2.4 Bounds set to the current research project

In the first year of the project only the author was working on the research sponsored by R.S.R.E. at Nottingham University. In order to provide as little restriction on the research as possible, and to avoid any potential problems with security classification, very little guidance was given to the research.

The research contract when continued by the author had been in existence for some 15 years, during which time research had largely been directed to problems of human visual perception. In the six month period immediately prior to beginning work the sponsors at R.S.R.E. Malvern changed the nature of research and Dr. A. Moy, the author's immediate predecessor on the contract, had started on initial research into the use of Sobel operators as an extension of his work on edge perception in the human eye. It was felt, however, that since the author had no experience in this field a fresh start should be made.

In the first instance it was felt that an understanding of the nature of pattern recognition could best be gained from a detailed study of a particular example of pattern recognition. Since this was an initial study it was decided to start with a preprocessing algorithm that would also provide a further insight into the nature of the pattern recognition problem. To this end and because Professor Beurle (the project sponsor at Nottingham University) has some significant experience in the subject, it was decided to make a study of the effects and uses of Fourier transforms on image data. The details and results of this research are given in chapter 3.

The equipment in use at this time was only that which was available through general University facilities, and this proved to be inadequate for the intensive computational requirements of the research. It was found that dedicated microcomputers available did not have sufficient memory for any but the smallest matrices, and that the departmental weekly time allocation on the University mainframes was insufficient for a full week of research using Fourier transforms. On one occasion the entire department CPU allocation for the week on the Computer Centre VAX was expended on a single Monday afternoon. For this reason an application was put to R.S.R.E. for additional funds to purchase a microcomputer for the project, and a UNIX based SAGE IV micro was purchased, with 2Mbytes of memory and 40Mbytes hard disk for storing images and programs.

R.S.R.E. also supplied a set of test data known as the 'Alabama Database', containing images of a plain in Alabama, U.S.A. with various tanks, jeeps and buses at known locations. This set of data was transferred from the tape to the University VAX, and from there to the SAGE micro. All of the Fourier studies using real data, as opposed to artificially generated boxes and lines, was performed using sections of this data.

The Alabama database provided a useful insight into the nature of types of data, that could be used. However, the definition of patterns within the image was very poor. The largest object was approximately 25 pixcels in total diameter, and blurring of approximately 3 to 4 pixels was present across the whole image. Also the images were taken using a thermal imager, thus the object patterns were not easily recognised as vehicles even when presented in an optimal way, the engine and wheels were particularly bright, and reflections of the sky from flat surfaces on the vehicles were particularly dark, thus it was very difficult to assess the performance or usefulness of any algorithms. The database would have been suitable for research into semantic pattern recognition, where the visual cues available over large regions of the image could be useful, however this would have required more images than were available, and more computing power (more memory in particular) than was possessed by the research project at that time. For this reason image capture apparatus was bought allowing simple visual images to be captured, and a database of suitable images was constructed.

During this time Mr. M.F.Daemi joined the research contract, and started to gather relevant information and references in a computerised database, and this has proved to be a significant asset to the development of the ideas discussed in this thesis.

With the new image capture apparatus, and a greater understanding of the Pattern Recognition problem provided by the first year of research, several foundations had been built for the next two years of study. Fourier Domain techniques had proved to be useful as a tool in other research approaches, but seemed unlikely to provide the final answer to pattern recognition, thus it was decided to look at pattern recognition from the perspective of shape recognition for the remaining research period. This was also hoped to compliment Mr. Daemi's study of Pattern recognition from the perspective of information theory. It was considered that the author's work on image capture and preprocessing would provide useful raw data for analysis of information, and that this analysis could in turn provide clues to the potential usefulness of recognition methods tried by the author. The work on shape extraction and contour descriptions is given in chapter 4 and the work done in assessing methods of recognition of these contours is described in chapter 5.

# Transforms

## 3.1 Objective

The principal reason for the study of Fourier and associated transform techniques made by the author was to gain an insight into the nature and properties of patterns as found in digital images. A secondary reason was to attempt to find a method of recognising patterns.

The use of Fourier transforms in pattern recognition is a long established field, thus it was thought unlikely that any significant new results would come to light due to research by the author in his first year of work. However, the Fourier transform has proved to be a useful technique for showing a different perspective of data, and a thorough knowledge of Fourier transform techniques was considered beneficial.

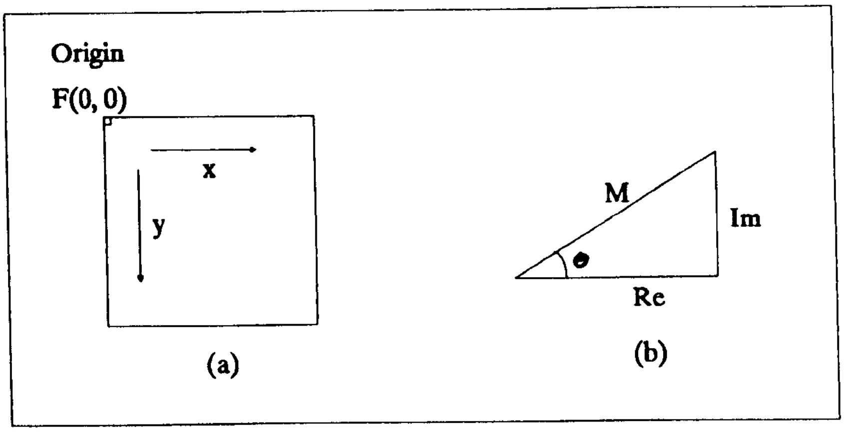
This having been stated, the research was not entirely indiscriminate, and was directed principally to determining any consistent patterns within images.

## 3.2 Fourier transforms

### 3.2.1 Representation

The result of Fourier transformation of an image is a matrix of vectors which may be represented using real and imaginary components, or using vector magnitude (modulus plane), and vector phase angle (phase plane). It is the latter description which has led to the most interesting results.

Figure 3.1 a) shows the location of the origin and transformed data points as described in the results in this thesis. Figure 3.1 b) shows the relationship used between the real (Re) and imaginary (Im) vector components, and modulus (M) and phase ($\theta$) components for individual vectors.




3.1 Data representations used in this thesis

The number of vectors produced by the transformation is equal to the number of original input points, however the algorithm assumes that the input values are complex vectors, whereas digitised images usually consist of a set of real components only. To overcome this difficulty all of the imaginary components of image matrices to be transformed are set to zero, thus only half of the data in the output matrix will be information. This shows up as a duplication of the data in element (x, y), which also appears in complex conjugate form in element (N-x, N-y).

### 3.2.2 Properties

The Fourier transform is the decomposition of a signal that varies with time or space, into its components of frequency or spatial frequency. In the case of a discrete Fourier transform it can be shown that only discrete frequencies need be used. This can be deduced from the continuous Fourier transform by considering that the discrete case may be represented by a sampled continuous waveform. Since the sampling process is repetitive, of theoretically infinitesimal duration at each sample point, and of fixed repetition rate, the continuous output matrix will be multiplied by the Fourier transform of this sampling process, giving a series of values at discrete frequencies. The maximum discrete frequency discernible with a Fourier transform is directly proportional to the number of samples taken per second, hence the number of output points is proportional to the number of samples.

An important property of the Discrete Fourier transform is its periodicity. If a signal is sampled at intervals T, the sampling frequency $f_s$ is 1/T, and this signal has a periodic spectrum that repeats at intervals of $f_s$. This effect is a consequence of the sampling, and occurs because a sine wave of frequency $f_o$ when sampled at frequency $f_s$ gives the same result as a sampling a sine wave of frequency ($f_o$ + i * $f_s$), where i is an integer. Intuitively, it can be seen that it does not matter how many whole sine wave excursions (i) take place between one sample and the next, it is only the partial sine waves due to $f_o$ that will be measured. For the purposes of the research the fact that frequency components outside the Fourier domain matrix may be considered as periodic repetitions of the Fourier plane, and therefore non-zero will be ignored.

A further property of Discrete Fourier transforms is that of aliasing. This states that a signal of frequency (i.$f_s$ f(x, y)) where i is an integer, will appear at the position of frequency f(x, y). Normally the bandwidth of the input signal is limited to $f_s/2$ to avoid the appearance of such signals, however in this case the bandwidth of the image signals is known to be limited, thus no filtering has been necessary.

## 3.3 Phase Plane Tilt

### 3.3.1 Introduction

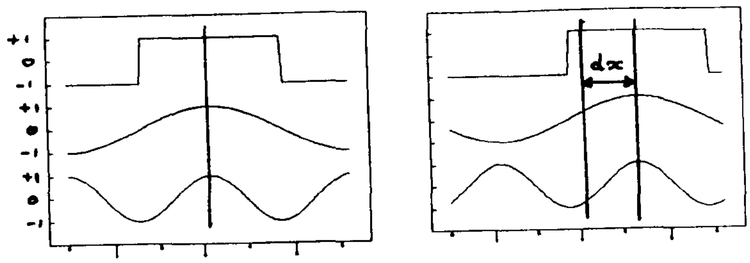
The first tests that were performed were on artificially generated images. A Gaussian bell centred on the origin was used to verify implementation of the algorithm, since this does not change shape under Fourier transformation. An 8 by 8 pixcel rectangle was used to verify the observation of ringing (see section 2.2.2.1, p.21), and further test the computer programs, including the translation of cosine and sine components into modulus and phase components. During this latter test a pattern consisting of low valued elements on diagonal lines repeated across the phase plane and further study was undertaken. An example of transforming the 8 by 8 rectangle in a 64 by 64 image plane shown in appendix A.1 is given in appendix A.2 and A.3. A further example of the phase plane tilt for an image with the same rectangle at a different location is given in appendix A.4.

It was found that for the Fourier transform of a simple object pattern on a noise free background in the image plane, the cell values on the modulus plane do not change with the position of the object pattern in the image. Since the Fourier transform is a linear transform, it is also true that the modulus plane will contain this same invariant pattern of data for any given object, irrespective of the amount of noise present. However, there will be other information related only to the noise, which will be superimposed on the pattern information, and this will tend to obscure the object pattern.

The results also show that the position of any prominent component of an image may be derived from the phase plane information. For a single point the phase difference ("step") between the phasor describing one frequency component and the phasor for the next higher frequency component is constant for all such pairs of frequencies. This occurs because the phase angle of the vector representing any frequency has a component that is directly proportional to the distance of the object pattern from the origin. The apparent exception to this occurs when the step from one phase component would take the next phase component above $360^{\circ}$. In this case the transform algorithm calculates the phase value less $360^{\circ}$.

### 3.3.2 Theory

The theory governing the phase step can be shown by referring to the definition of the Fourier transform. In figure 3.2 a) a simple one dimensional pattern is shown. Also shown are the first two Fourier harmonics.




3.2 Effects of pattern movement in the Fourier domain

Since the object is centred on the origin, and is symmetrical about the origin, the phase of both frequency components is zero. If, however, the object is shifted by a distance dx to the right as shown in figure 3.2 b), the Fourier components will also be moved to the right, and this is represented by a change in the phase of the nth frequency $\varphi(n)$ such that:-

$$\varphi(n)=\frac{dx}{\omega(n)}\times 2\pi$$

where $\omega(n)$ is the wavelength of frequency n. We also have that

$$\omega(n)=\frac{\omega(1)}{n}$$

Thus the phase change $\varphi(n)$ is given by:-

$$\varphi(n)=\frac{dx}{\omega(1)}\times n\times 2\pi$$

Thus $dx/\omega(1)$ is the per unit distance across the frame, or the number of pixcels to distance dx from the origin, divided by the frame width. It can be seen that since the phase change is proportional to the harmonic number of the component, the phase difference between two adjacent components will be proportional to the distance moved across the frame, since:-

$$d\varphi = \varphi(n + 1)\times 2\pi - \varphi(n)\times 2\pi$$

$$= \frac{dx}{\omega(1)}\times(n + 1)\times 2\pi - \frac{dx}{\omega(1)}\times n\times 2\pi$$

$$= \frac{dx}{\omega(1)}\times(n + 1 - n)\times 2\pi$$

$$= \frac{dx}{\omega(1)}\times 2\pi$$

This means that the phase plane is "tilted", that is, it has a constant gradient which may be detected and possibly removed, thereby shifting the object in the frame to the origin.

This simple relation is only true for symmetrical objects where the average of the cells that compose the object have the same effect as for a single point object. Non-symmetrical objects will have a component of phase angle due to their asymmetry that is not constant for all spatial frequencies, and this will tend to distort the tilt. However, when using real images these asymmetric components will normally be independent of frequency, thus they will tend to cancel if taken over a sufficient range of frequency components.

It is also clear that the deviation in phase angle due to asymmetry will be relatively large for high frequency components of the transformed image, when compared with those of the low frequency components. This occurs because the high frequency components are influenced by a large number of small variations, rather than the small number of significant influences on the low frequency components. Thus low frequency components, of a wavelength comparable with the width of object patterns in the image will be largely influenced by those object patterns, whilst high frequency components will be influenced by small details on the object.

In practical images, object patterns are generally brighter near the centre. The low frequency components may therefore be used to determine object location, and if required a constant tilt may be removed from the phase plane to place an object at a known position in the image plane. Very large object patterns in images with a significant degree of asymmetry will cause the tilt to be influenced by features inside the object, and these types of image should not be used with this algorithm.

### 3.3.3 Phase plane tilt detection results

The phase plane tilt detection algorithm works by producing the histogram of frequency of occurrence of step, versus step, and determining which step occurs most often. The results for a bright rectangular pattern on a noise free background are given in Appendix A.6 a). It is clear from this that there is no difficulty in determining the best step for this simple pattern, and the result of removing this step from the entire phase plane is to centre the rectangle on the origin.

The histograms in appendix A.6 b) and c) are for practical images taken from the Alabama database. The histogram in appendix A.6 c) is a thresholded version of that in A.6 b). The detail of the image in appendix A.6 c) is given in appendix A.5 a) and b).

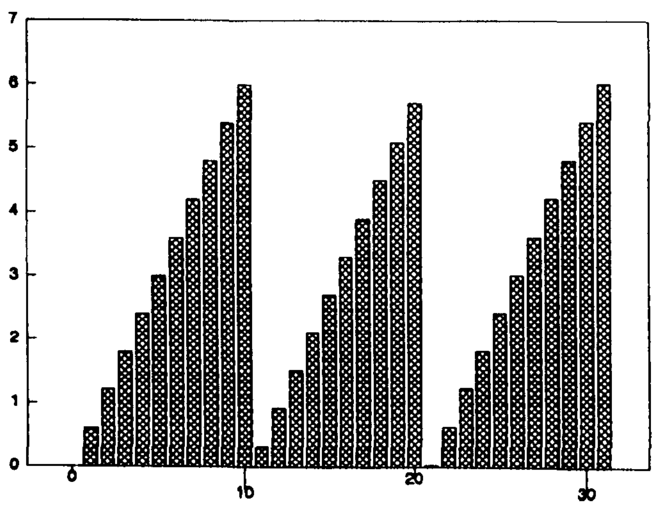
From these histograms it is clear that the noise imposed by real images is significant but not overwhelming.

The results of removing the phase plane tilt that is indicated by these histograms were disappointing. An example of removing the tilt for the image given in appendix A.5 (step histogram shown in appendix A.6 c)), is given in appendix A.7. The histogram itself required some smoothing to obtain a reliable result, however, the result obtained was influenced more by the edges of the image than by the object. As has already been indicated, the Fourier transform is repetitive, thus the transform is constructed as though the right hand edge were immediately adjacent to the left hand edge, and this will be the dominant feature in the phase step histogram.

At this time it was felt that the study had proceeded sufficiently to have achieved the objective of understanding the nature of noise in images, and some useful experience had been gained in the development of computerised routines for algorithm modelling. It was noted, however, that using the phase plane to determine the location of an object could only work for a single pattern in an image, even under ideal circumstances of noise and background grey level variation. For this reason a new line of research was started, and no further study of phase plane tilt took place.

### 3.3.4 Fourier transformation of phase information

As outlined in the previous section, the phase angle of frequency domain vectors in a Fourier transformed image gradually steps up in value across the phase plane, until it reaches $2\pi$ radians, whereupon the phase angle value at the next higher frequency has returned to just over 0, having gone through a full circle. If a cross section of the phase plane is considered, it can be seen that the angles will follow a repetitive ramp function such as is shown in figure 3.3.




3.3 Tilt in the Fourier phase plane

If this wave shape is subject to a further process of Fourier transformation, it would be possible to determine the frequency corresponding to the repetition rate of the ramp function, and this is related to the tilt of the phase plane, and hence the position of an object by a simple relationship, which may be determined as follows.

Let dx be the distance of the object pattern from the origin. It has been shown that the phase step $d\varphi$ between adjacent frequencies is

$$d\varphi = \frac{dx}{\omega(1)}$$

The wavelength ($\omega_s$) in pixcels across the frame is given by

$$\omega_s = \frac{1}{d\varphi}$$

$$= \frac{\omega(1)}{dx}$$

Thus the frequency ($f_s$) may be found from the relation

$$\omega_s \times f_s = \omega(1) \times f(1)$$

Thus

$$f_s = dx \times f(1)$$

This represents a frequency which appears approximately in the same position within the frame as the original input object pattern. The effect will only work if the step in the phase angles is constant for a significant number of steps, this is true for small sharply defined edges as found around a object pattern, thus it is expected that this technique will highlight the location of an object, providing it is symmetrical.

A further consideration is the case of a object pattern which contains Fourier components with negative coefficients, either because the object pattern itself is a dark object on a light background, or because the object pattern contains some comparatively dark features. In this case the phase angles will be displaced by $\pi$ from the expected position, and may destructively interfere with each other. However the coefficients of the second harmonic will interfere additively. For this reason it is supposed that the second harmonic may be a better indication of the location of an object pattern than the first. The position of this harmonic will be at twice the distance from the origin as the first harmonic, that is

$$2\times f_s=(2\times dx)\times f(1)$$

## 3.3.5 Walsh-Hadamard transforms

### 3.3.5.1 Introduction

The similarities between Fourier transforms and Walsh-Hadamard transforms are not fully apparent on first inspection, however it is noted that the methods of construction are very similar. The Walsh-Hadamard transform is similar in nature to the Fourier transform if only the sign of the cosine wave is used as a multiplication factor during summation. Also the spectrum produced by the Walsh-Hadamard transform may be very similar to that of the Fourier transform under certain circumstances.

The most notable difference is that Walsh-Hadamard transforms do not use complex representation, only scalar representation. This is achieved by interleaving the equivalent to the cosine components and the sine components throughout the matrix. If the output "sequencies" (analogous to frequencies) are considered it is found that sequency 0 corresponds to frequency 0 in the Fourier domain, sequency 1 corresponds to a square wave version of the fundamental sine component, and sequency 3 corresponds to the Fourier cosine component. Sequency pairs 2 and 6 correspond to the second harmonic of the Fourier transforms in the same way. Other sequencies are composed of harmonics which do not conveniently fit in one frame width (eg. third harmonic), and are thus apparently random in distribution.

#### 3.3.5.2 Algorithm

Although the Walsh-Hadamard transform may be constructed in a very similar way to a Fourier transform, this is not the most efficient method. The Fourier transform requires that each element is multiplied by a phase factor before summation for each output element, and the Fast Fourier transform makes use of the fact that each element is frequently multiplied by one factor for summation into many separate output elements. The order of dealing with elements is chosen such that the phase product is found only once and added to the necessary output elements in several consecutive program steps, this part of the algorithm is generally known as the "Butterfly".

Since no multiplication need take place in the Walsh-Hadamard transform it is possible to simplify the process by building the addition and subtraction implied by the factors of +1 and -1 into the program itself, instead of obtaining them from a lookup table. By this means the process time may be reduced. The order of output element calculation is the same as for the Fast Fourier transform.

A routine was developed by the author which would perform Walsh-Hadamard transforms in optimal order, overcoming the problem of matrix reordering which is involved in the transformation. The concepts behind the routine are very similar to those governing the Fast Fourier transform. The order of transformation is such that only a minimum number of additions and subtractions are performed. However, since there is no multiplication factor other than plus or minus one used in this transform, the addition and subtraction are performed within the routine, giving a further time saving overall.

#### 3.3.5.3 Walsh-Hadamard transform Test Results

The most significant problem with the Walsh-Hadamard transform is the dependence of the output spectrum upon the position of an object in the frame. In Fourier transforms it has been seen that all of the information relating to the position of features is contained within the phase plane, however there is no equivalent plane in Walsh-Hadamard transforms. Therefore a test was performed to transform a one dimensional pulse and compare the results with the equivalent Fourier transform.

A rectangular pulse of value 256 and width 21 elements was placed in an array and Fourier transformed. The modulus part was found and displayed on a printout. Alongside were displayed the results of Walsh-Hadamard transforming similar pulses which started at elements 1, 26, 51, 76, 101, 126, 151, 176, 201 and 226, that is, at intervals of 25 cells giving a spread across the range of possible positions. A portion of the results of this test is given in appendix A.10.

In all cases the results show high values at the low frequency/sequency positions and lower values at the high sequency positions. All of the values from the Walsh-Hadamard transform were integral multiples of 256, as a consequence of all input values being 256 and there being no sub-division of this value within the algorithm. It was also noted that the values for any particular sequency in the Walsh-Hadamard transform output varied greatly as the pulse was moved across the array, and that patterns were formed in the output matrices which are not easily related to each other. The region around the origin of a two dimensional Walsh transform result for an 8 by 8 square on a 64 by 64 image plane is given in appendix A.9 a). A similar image with the square in a different place in the image plane is given in appendix A.9 b). It is clear from these images that the transformed result varies significantly with object position. For this reason the Walsh transform was not pursued as a fast method of transformation.

### 3.3.6 Combining Fourier planes from different images

It had been noted during the research that the modulus planes of various images did not show any significant difference on first inspection. All of the digitised images tended to be a peak around the origin that dropped in an approximately bell shaped curve to small values at some distance from the origin. The distance is determined by the size of the object that has been transformed. A test was therefore tried that combined the modulus plane of a rectangle with the phase plane taken from the Fourier transform of one of the Alabama database images. The results in Appendix A.8 show a tendency to highlight the regions in the image that most conform to the pattern in the modulus plane.

It is supposed that if this line of research were pursued a novel method of thresholding images may be found, that could be particularly immune to noise. However, due to the constraints of time, and other associated research projects this was not continued.

### 3.3.7 Conclusion to work on Fourier transforms

The objective of determining the nature of image data that may be used for pattern recognition has been adequately met, however, the use of Fourier transforms as investigated in the research has proved less fruitful.

The algorithm for locating a symmetric object using the tilt of the phase plane fails; not because of noise, but due to the nature of the background to the images to be transformed. The method is also fundamentally flawed, in that it cannot deal with more than one object in the field of view.

Taking the Fourier transform of the phase information is a potentially useful method of object location. Results show easily identifiable peaks around the location of the object, however, it is not clear that this approach is any better than simply thresholding the original image.

The use of Walsh transforms has shown that although they are particularly efficient in terms of speed of processing, the drawbacks of variation with shift means that they do not lend themselves to any of the approaches for image transformation tried by the author.

Using the modulus plane from a noise free square and the phase plane from a real image shows some promising results. The results show a tendency to highlight the parts of the image that most conform to the size of the rectangle, whilst retaining the shape information virtually intact. It is thought that with further investigation, this approach may give a method of thresholding an image with a great deal of noise immunity.

# Contour Following

In the second part of the research period the author chose to investigate patterns in terms of their shape, in an attempt to find a method of describing shapes that was both economical in terms of data, whilst retaining as much shape information as possible. The approach chosen was to describe the object in terms of the path followed about its periphery, and base further recognition algorithms upon this.

## 4.1 The information content of shape

The reason for assuming that the shape is relevant to recognition comes from information theory. It is clear that in a fully utilised data array all data elements can carry the same quantity of information, that is each data element is as important as any of the others. If, however, it is known that a particular data element always has the same value then the data element carries no information additional to that which is already known. Similarly, if the value of a data element is constrained to be within a specific sub-range of all possible values then the amount of information is correspondingly reduced. This holds for images that have been reversibly transformed in some way, thus knowledge about the gradient and other measurable parameters of regions may reduce the amount of information held in an image.

In many cases the objects seen within an image consist of regions of small variation in value, bounded by regions at a different grey level. The regions within the bounds are therefore constrained, and carry relatively small amounts of information, whereas the boundary (i.e. shape) data elements may take any value, and thus carry the majority of the information.

The maximum amount of information that can be obtained from any static data is defined by the number of binary digits that are required to describe the data. From this it is known that the maximum amount of information that can be obtained from a 32 by 32 pixcel image with 8 bits of information per pixcel is 8192 bits. However, it will be shown in chapter 5 that for objects described in terms of their peripheral paths alone the amount of data can be reduced to 10 bits, and yet still retain much of the information associated with pattern identity, position and orientation.

## 4.2 Use of Assumptions

### 4.2.1 Introduction

It is possible to include assumptions in systems which evaluate information by including knowledge within the system that is not intrinsic to the information presented by the data, for example by limiting the range of valid values for a data element. The assumptions built in to the recognition algorithm restrict the number of choices of result, i.e. they reduce the "channel capacity" of the algorithm, and thereby increase the efficiency of the algorithm in recognising useful information.

### 4.2.2 Assumptions used

In the research the author has used a number of assumptions about the image data to simplify the problem. Without this the task of analysis would be virtually impossible (i.e. all $2^{8192}$ images would be equally valid). The assumptions used, however, are not unrealistic, and it is hoped that from the basis of these assumptions more complex images may be analysed using similar techniques by improving the segmentation method. Since literature on the study of segmentation is now widespread a number of approaches to advancing the techniques suggest themselves for future work.

The a priori knowledge that is additional to the information in the images used is as follows:-

*i)* The "background" is "dark", i.e. the object pattern is completely surrounded by a region of pixcels that will always be less than any chosen object pattern to background threshold value.

*ii)* The "object" is "light". The pixcels encompassed completely by the object are saturated with light, and register a value that is the maximum that the digitising system is capable of outputting. This value is numerically $236_{10}$ ($EC_{16}$).

*iii)* The object as seen by the camera has sharply defined edges, and is well focussed.

*iv)* The object does not overlap or touch the edges of the field of view.

*v)* The object is a three dimensional object, not symmetrical about all axes.

From practical knowledge of objects and the images used it is also assumed that:-

*vi)* There is no "special" place on the periphery of the object. Thus the point at which the contour is first traced is not used in the recognition of the object. This arises directly from the assertion that the algorithm should be independent of the grid used to digitise the image.

*vii)* All images used contain only one object.

### 4.2.3 Effect of assumptions used

It is clear from i) and iv) that if a point on the object is known, then the closed set of zero valued elements immediately adjacent to the object serve to define the background, and that all pixcels outside this region do not give any additional information. Appendix B.1 shows the complete set of data points for a typical image from the captured set. Appendix B.2 shows the same data after removal of the zero valued elements described above.

By similar logic the largest area of peak values that is within the region described above is already given by ii) and therefore also supplies no additional information. Appendix B.3 shows the effect of removing these data elements.

The exceptions to the foregoing are pixcel boundaries that have a zero valued pixcel on one side, and a peak valued pixcel on the other. In these cases both pixcels are required to mark the location of the boundary.

The information given by iii) is useful in determining the number of pixcels around the periphery of the object that must be considered. It is known by observation of the characteristic of noise introduced on the background, and on the peak regions of images, that there is a small amount of noise introduced into the images by the camera and digitiser. This does not generally amount to an error in the illumination intensity value of more than 4 parts in 236. It is also known from the algorithm used to reduce noise in the digitised images that each pixcel in the image represents the mean value of a light detector of approximately uniform sensitivity over the whole area of the pixcel. This is given by the use of averaging over a rectangular array of 15 photocells to obtain each pixcel value. Therefore it is shown that the minimum width of the edge of the object as found on the digitised image will be negligible, and the maximum apparent width will be two pixcels, horizontally, vertically or diagonally.

The assumption in vii) has been used to simplify the segmentation, however the algorithm used could cope with many non-overlapping objects within one image, each object found would be treated as independent of any others in the image.

The diagram in appendix B.4 illustrates for a sample image the amount of actual information that can be obtained from the image after the data that supplies no additional information has been discarded.

To be sure that no information has been lost, it must be possible to rebuild the original image from the information available in the image that has been simplified by use of assumptions, taken with the information about the image that is built into the algorithm. In the case illustrated in appendix B.4 it is possible to rebuild the image, and thus in using the assumptions above no information has been lost. In several of the other images from the set there are non-zero points on the background. These represent lost data in strict terms, however knowledge of the original digitised scene indicates that the data is due to noise within the camera or digitiser, and therefore no investigation into methods of eliminating this data loss has been carried out.

It can be seen from the diagram in appendix B.4 that the information given by the selected object is much less than the potential information that could be given using all of the available data points. This is a most desirable result since at best the length of time taken to recognise an object is directly proportional to the number of binary digits needed to represent the object. Thus the assumptions used have significantly improved the recognition process for this class of object.

### 4.2.4 An example of the effect of the assumptions

The diagrams in appendix B show the effect upon an image of the assumptions built in and the algorithm used. Appendix B.1 shows the original data in its entirety. The data is given as two digit hexadecimal numbers for convenience of representation. In appendix B.2 the zero valued pixcels that are entirely outside the object and not touching any non zero pixcels are shown removed. This is a realisation of item i) of the assumptions. It can be seen from this that much of the peripheral shape of the object can be determined simply from the pixcels that have zero value and are touching non zero pixcels.

The diagram in appendix B.3 shows the effect of removing the peak valued pixcels, which are already known about from item ii) of the assumptions. The remaining pixcels carry the shape information of the object. It should be noted, however, that there are also pixcels remaining inside the pattern that are not part of the edge of the pattern. These represent the shadow cast by the aeroplane fuselage over the wing, and therefore appear only on one side of the fuselage. Consideration was given to producing more uniform digitised images when this shadowing was first identified, however the segmentation used has proved adequate for the purpose of shape recognition, and further tests of segmentation techniques could be carried out on the same data at a later date if the shadowing is retained. The lines a-a', b-b', c-c' and d-d' show the lines of the cross-sections illustrated in shown in figure 4.1 a) to d).

The diagram in appendix B.4 is illustrative of the shape of the aircraft. It is similar to appendix B.3, but has had all zero valued pixcels removed. The engine on the port wing, and structure of the tailplane can clearly be seen in this figure, and this gives an indication of the resolution that can be achieved on the pixcel level.

The diagram in appendix B.5 shows the data items that have actually been used to segment the image. The segmentation method used is an enhanced thresholding operation, and the pixcels shown are those pixcels that are immediately adjacent to the contour given by the threshold value. It should be noted that the threshold contour follows the line of the grid between the pixcels, and no lines running across the diagonal of pixcels are included.

It can be seen that the segmentation fails to include 50 of the 212 pixcels from the image, and includes 4 other pixcels that are not given in the ideal image. Of the 50 pixcels 31 have the value 0 and are adjacent to pixcels lower than the threshold value, 12 have a value greater than 190 ($BE_{16}$) and are near to the central sections of the aeroplane, 6 are within the shadow region of the port wing and the remaining pixcel has value 7 and is near the tip of the port wing. This single pixcel is separated from the object by a pixcel that is also lower than the threshold value.

Theory predicts the edges of the aircraft as seen by the digitiser will not be ideally sharp in most circumstances, because the edges of the object will rarely pass along the edges of the scanning grid. This can be seen to be the case in practice. The majority of differences between the segmented image and the ideal segmentation occur where zero valued pixcels do not touch periphery, because pixcels with values less than the threshold value form part of the periphery. The ideal segmentation is a segmentation that takes non-zero cells as belonging to the object, and zero valued cells as part of the background. That the ideal edge is occasionally found has been shown by the inclusion of 4 peak value pixcels adjacent to the edge.

For these reasons it is considered that although the segmentation method does not follow the theoretical ideal, it is nevertheless a good approximation for the limited sub-set of images that have been chosen for study.

The final figure of the set (appendix B.6) shows the coarse contour that is given by the segmentation. The approximate outline of the aeroplane can be seen from this diagram, however the problems inherent in using such a simple segmentation technique are also clear where the "steps" occur instead of the straight diagonal lines of the original, for example from the starboard engine to the starboard wing tip. This is the problem already indicated for chain codes in section .

## 4.3 The practical pre-processing algorithm

The algorithm used to obtain the diagram in appendix B.6 is well described by the foregoing paragraphs, however, a few details are noted here.

### 4.3.1 Image Capture

#### 4.3.1.1 Apparatus

The objects used for study were a toy aeroplane, a toy car, a wooden ball, and a number of solid geometric shapes constructed from paper, e.g. a cube and a cone. For the purposes of testing the contour following algorithm only the aeroplane and the car were used, though all objects were used in testing the segmentation techniques.

The aeroplane was painted white over all of its surface with the exception of a set of black plastic wheels. The wheels were attached using a large metal stud, and reflections from this stud initially presented problems in segmentation. The principle cause of the problem was the automatic sensitivity control within the CCTV camera. If a particularly bright spot on the object was found then the camera would dim the overall image to compensate. This in turn causes shadows to deepen, and thus the range of possible threshold values becomes smaller. A discussion of the problem of shadows on the aircraft is given later in this section.

The problems were overcome firstly by carefully adjusting the lens aperture control to give maximum contrast, thus saturating most illuminated pixcels, and secondly by carefully selecting the threshold values by which images were segmented. Some consideration was given to simply cutting the wheels from the aeroplane, however, it was felt that by making the image too easy to segment could cause the author to overlook some important and unforseen aspect of the recognition.

The aeroplane was mounted on a turntable, which was in turn held by a clamp stand. The camera was mounted pointing directly at the centre of the turntable, thus any view of the aircraft from above, front, back or side could be presented to the camera.

Illumination was provided by natural daylight from the window. This source was used since it was found that other more controlled sources such as lamps cast sharp shadows and irregular patterns over the aeroplane even when reflected from white card acting as a diffuser.

The camera itself was a low cost CCTV camera with a built-in automatic sensitivity adjuster and a diaphragm lens aperture control. The output signal from the camera is sent to a low cost digital sampling circuit controlled by a BBC micro computer.

#### 4.3.1.2 Sampling hardware

The digital sampling was performed using very low cost hardware, and it is felt that this section of the process could easily be improved. The sampler was controlled by a BBC micro, and was thus relatively easy to control once the software to do so had been written. However, because the communication between the sampler and micro was also of very low cost, it was not completely reliable. Loss of frame synchronisation, and of data points if the BBC micro was interrupted, was not uncommon. For this reason some effort was spent in ensuring that the images captured were useable, and the control software was written to this end.

### 4.3.2 Control software

The control software was written to request and accept a set of data points, one from each frame line, for each frame scan of a scene. The process of obtaining a column of light intensity values was repeated a total of three times, and then the sum of values in 3 by 5 sample points was taken and stored as the result for a single output pixcel.

The choice of this scheme was based upon a number of factors. Firstly the BBC micro has only 32k bytes of random access memory. To store all possible sample points would require 67k of memory. To store only the 32*32*15 points used requires 15k of memory, but including the program, operating system and screen display memory the total memory requirement is in excess of the 32k available. The scheme used requires only 1.5k of memory for data storage, and is therefore feasible.

The use of a 5 by 3 set of sample points to produce an output pixcel improves the signal to noise ratio of the resultant data byte by nearly 2 bits. It also enlarges the region over which the scene is scanned, and improves the aspect ratio of the image to approximately 1.05:1 from 3.15:5. This improvement is most important for experiments where the object rotates with respect to the camera.

### 4.3.3 Analysis software

Once the values of the output pixcels had been determined and stored for the column each value was used to present a screen display of the image. The 6 most significant bits of the value were used as an index to a look-up table of display dots, index 0 produced a pixcel display output with no screen dots, index 1 produced 1 dot etc. up to index 63. The screen display dots were placed onto the screen in the correct position relative to the output matrix, thus as the scene was digitised an image was produced of the output matrix.

This feature proved invaluable for a number of reasons. Firstly it facilitated the location and sizing of the object within the frame, secondly it assisted in the correct setting of the lens aperture. This was made difficult since the automatic sensitivity feature of the camera tended to negate any changes made using the aperture control, and the automatic contrast control in the monitor would not allow any changes to be seen. Thirdly the display showed that the system was working correctly. A loss of line or frame sync after startup was a regular occurrence, probably due to inadequately shielded connections. This was not detected by the software used, due to time constraints imposed by the line sync period, and a simple visual check ensured that the system was running. Similarly the digitiser would occasionally produce a number of frames that were out of synchronization with respect to the rest of the image, giving a narrow vertical band of pixcels that were shifted by four or five pixcel heights. Ensuring that the image captured was not affected by this extraneous noise was most useful.

Consideration was given to disabling the automatic sensitivity control of the camera. However, the camera was under warranty, and the time scale of obtaining money or equipment (usually several months of waiting) from R.S.R.E. should the camera be broken was felt to be good reason for leaving it intact.

The final reason for requiring some form of image display was that in the early stages of research it was particularly time consuming to convert images from the BBC micro-computer floppy disk format to other systems. This problem was solved by the purchase of a KERMIT communication and terminal emulation ROM, however the effort involved in transferring the images at the start of the research meant that it was important to ensure that the data being transferred was usable.

The summation of data points was performed using two byte integer arithmetic as readily provided by the 6502 processor within the BBC. The product of the maximum value of a single sample point ($111111_2$) and the number of sampled points in one 3 by 5 point pixcel is $361_{16}$. This cannot conveniently fit within one byte, and it is shifted right by 2 bits to give a peak value of $EC_{16}$ ($236_{10}$). It is this value that appears as the peak pixcel value in the examples given in appendix B.

The image scan process was made to repeat continuously to facilitate setup of scenes, however, a keystroke caused the scanning to stop at the end of the current scan. The image captured could then be analysed by a short BASIC program to determine the distribution of pixcel values, and could also be saved as a binary image to disk for subsequent transfer to the SAGE computer.

## 4.4 Pre-processing

The algorithm used to obtain the coarse contour around the principle object in an image consists of the stages of gradient removal, thresholding, contour description and best object evaluation.

### 4.4.1 Gradient removal

Due to imperfections in the manufacture of the CCTV camera the background of the images captured were not always perfectly dark. If too much light was used to illuminate the object then the material used as a background gave some reflection. If too little light was used then the sensitivity control within the camera would cause pixcels to have non zero values even if they were perfectly dark. For example if the lens cap were on the camera then most pixcels would have a non zero value, and the average value of pixcels would be between 8 and 12. This effect was not uniform over the whole of the digitised region, the top left hand corner being raised by up to 20, whilst the bottom right hand corner was usually unaltered giving a gradient to the whole image.

For these reasons the image was filtered using a gradient removing program. The program calculates the average value of the left and right hand columns of pixcels separately, and applies a reversing gradient as appropriate to make the difference in the two values close to zero. The program then repeats for the top and bottom values of pixcels to remove any vertical gradient.

The removal of gradient using this approach is not ideal since in practice the original slope is not uniform, and does not affect pixcels that were initially saturated with light, thus compensations on the object apply an amount of image degradation. However the result obtained was satisfactory.

To improve this section of the algorithm it is recommended that a local gradient transform is used since this would overcome many of the problems associated with the addition of absolute values to the image, such as are imposed by a light background, or non uniform camera sensitivity. The difficulty of this approach for our purpose is that the contours produced are not unconditionally closed.

### 4.4.2 Thresholding

The process of thresholding has been used to locate a closed contour around an object that best follows the edges of the object. As has already been seen this reasonably approximates to finding the non zero edges to the object. However, it has also been noted that there is noise present on the image, and also features of approximately one pixel in width, thus using a threshold value of between 0 and 1 is not reliable.

The first threshold value tried was simply the mean value of all pixels in the image, however this tended to give too low a threshold due to the relatively large number of zero valued pixels. This therefore lost any small features on the periphery of the object pattern as they were swamped by noise introduced by the camera.

The second method used was to take the mean value of all non zero pixels. This gave too high a threshold value since it was greater than the pixel values of shadows on the wings of many of the images, thus a number of contours were produced of aircraft with missing wings, which were located as separate objects.

The method that was used was to take one third of the mean value of all non zero pixels, this produced a satisfactory contour in all cases except image 00/060. The image that failed to give a suitable contour using this algorithm has particularly bright reflections from the aeroplane wheel studs. In this case the solution used was a trial-and-error method. Even with the best (subjective) contour the resulting object does not correlate well with any other, and can be taken as an example of a potential pitfall in this method of pattern recognition.

This approach to finding the rule governing best threshold value used has been unsatisfactory. Part of the problem lies in the need to have recognized the type of object being segmented at this very early stage of processing. Other solutions are to use the rate of change of pixcels approach as previously mentioned, or to use an iterative solution whereby the threshold value used is modified after some further initial processing has taken place. An adjunct to this would be to base the description of objects upon multiple contours taken at different threshold values. Clearly in this case the amount of processing required would increase greatly with the number of contours, and it is not clear that the benefits gained in the majority of cases would warrant the extra processing involved.

### 4.4.3 Contour description

Having determined the method of finding a threshold, and evaluated a threshold value for a given image, a closed contour can be produced around any objects within the image. At this stage there is no means of knowing how many objects are apparent in the image, thus all closed contours must be extracted and evaluated. It is known from assuption vii) that for this study there will only be one recognisable object within the image, however there may be more than one closed contour due to noise in the original digitised image.

A matrix pair is constructed that records all pixcel edges that lie between a pixcel above the threshold value, and one below the value. For practical reasons two matrices are held, one records the horizontal boundaries and has 31 * 32 data elements, the other records vertical boundaries and has 32 * 31 elements. The edges of the image always show as an object boundary to ensure that all contours are closed, however other matrix elements are only set if they are actually on a boundary, and are reset when the boundary is noted as belonging to a specific object. By this means no boundary is made to belong simultaneously to two separate objects. It should also be noted that at this stage the information about specific pixcel values is not used, thus a dark object on a light background and a light object on a dark background will be treated identically.

The algorithm next repeatedly searches row by row for a vertical boundary. If none is found then the contour extraction section is finished and any contours located can be processed further. If a vertical boundary is found the routine searches for connecting boundaries in a clockwise sense about a light object. Left turns are favoured above straights or right turns around the boundary to ensure that the longest possible connected contour is found. When no more connections can be made the contour must be a closed one, and the routine begins searching for another starting position.

### 4.4.4 Best Contour Selection

The contours are held as linked lists of boundary position records attached to linked header records. For each boundary element traversed a counter in the header record is incremented. If the boundary element is on the edge of the image then the counter in the header record is decreased by 10, this has proved sufficient to ensure that any small object touching the edge will be disregarded, and that any large object that just touches the edge of the image may nevertheless be selected. By this method the largest object wholly within the image and not touching, or only just touching, the edge of the image is selected. A high valued pixcel due to noise at the top left hand corner of the image would give a contour that encompassed the entire image frame, and thus be comparatively long, however, this is not selected as the best contour since a great many of the elements will decrease the recorded length rather than increase it. Note that the algorithm could easily be altered to include several objects for further analysis, but assumption vii) makes this unnecessary at this stage.

The boundary described by the interstices between pixcels when the value is greater than or equal to the threshold value on one side, and less than the value on the other side is not an adequate description of an object if a contour description is to be used. A number of problems can be identified. Firstly a line drawn at 45° relative to the digitiser grid will be $\sqrt{2}$ times longer in total length than a line drawn parallel to the axes. This is the same problem as has been identified with chain codes in chapter p.42.

The second problem is the significant loss of resolution in the resultant image. Information is available which gives an indication of the exact position of the object boundary from the values of the pixcels adjacent to the lines. In the ideal black/white image where the boundary does not follow the grid axes the value of each digitised pixcel is directly proportional to the amount of its area that is illuminated by the object. By using information from adjacent pixcels the point at which the best contour crosses the centre line of the pixcel can be found.

The effect of this can be seen in a practical object from the charts shown in figure 4.1. The charts represent a graph of illumination versus distance for each of the lines shown on the figure in appendix B.3. Part of the effect seen is due to shadow and poor focus, however it is clearly seen that even at the most sharply defined edges there are pixcels that lie between zero and the maximum intensity value. By selecting an appropriate threshold value it is possible to chose the steepest part of the rise for an image over as much of the boundary as possible, and thus have the most exactly defined edge possible.

### 4.4.5 Interpolation

Assuming that the object boundary passing through each pixcel is approximately straight allows a more precise definition of the position of the boundary. Figure 4.3 shows adjacent pixcels of digitised value A and B. Assuming that the intensity varies linearly over the whole pixcel the centre point of the pixcel will have the exact value measured. Shown below the pixcels in figure 4.2 is a table of actual intensity at specific points between the pixcel centres. The intensity crosses the threshold intensity value T at point t. From the theorem of similar triangles we have that:-

$$\frac{t-b}{a-t} = \frac{T-B}{A-T}$$

Let $\frac{T-B}{A-T}$ be Q

Then

$$t-b = (a-t)\,Q$$

$$t\,(1+Q) = Q\,a + b$$

If $a-b$, the width of a pixcel, is denoted as length x, then

$$a-b = x$$

$$a = x + b$$

$$t\,(1+Q) = Q\,(x+b) + b$$

$$= x\,Q + Q\,b + b$$

$$t = \frac{x\,Q + Q\,b + b}{1+Q}$$

$$= \frac{x\,Q + (1+Q)\,b}{1+Q}$$

$$= \frac{x\,Q}{1+Q} + b$$

$$= \frac{\frac{x\,(T-B)}{A-T}}{1+\frac{(T-B)}{A-T}} + b$$

$$= \frac{x\,(t-B)}{A-T+T-B} + b$$

$$= x\,\frac{(T-B)}{A-B} + b$$

From this we can deduce a closer approximation to the position of a point on the boundary. The assumptions that the centre of a pixcel has approximately the mean value of the whole pixcel means that the point determined lies on a line connecting the pixcel centres. This assumption is not justified for all points, however it is correct for regions where the boundary is not sharply curved, and it is these regions where it is most important that the exact position of the boundary is found.

## 4.5 Data normalising

### 4.5.1 Image Selection

In order to test the algorithms with an adequate set of data the model aircraft was viewed and digitised from a large number of angles. The first views were all taken from directly above the aircraft, perpendicular to the plane of the wings; this set is denoted by 90/nnn in any diagrams. The aircraft was digitised with the nose pointing to the left of the image giving sample 90/000. The turntable was then rotated through $15^{\circ}$ to give image 90/015. The process was repeated for each $15^{\circ}$ interval, giving a total of 24 images. The turntable was then tilted through $15^{\circ}$ away from the port wing, and a second set of images was digitised denoted as 75/000 through to 75/345. The process was repeated for elevations of 60,45,30,15 and 0 degrees, thus a total of 168 views of the aircraft evenly distributed in a hemisphere above the aircraft were available for test purposes.

### 4.5.2 Angular Description

The points located on the contour form a better approximation to the original image. Further improvement may be gained by careful filtering of the resulting image to straighten the long edges of the object, however this would require very detailed knowledge of the position of the edges in the original image which is not available from the digitisation used.

Problems associated with the points describing the contour are that they are not evenly spaced around the object boundary, the number of points may vary depending upon the apparent size of the object and the point coordinates are in terms of the absolute position within the matrix.

The apparent size of the object to be recognized is not necessary for shape recognition, also pattern matching is greatly facilitated if the size of the pattern and template are the same. For this reason the number of data items describing the contour is normalised.

To remove the absolute location and orientation information the absolute position information is converted to relative vector information. Each vector should have the same length and thus the lines connecting the known points are first divided into a number of separate sections. In this case one eighth of a pixcel width has been used as the section length, with any remainder rounded to one whole length or zero. The angle between each section and the next is then recorded, thus the absolute orientation information is extracted. The absolute angle of the starting vector is recorded to retain all available information.

### 4.5.3 Size Normalisation

The number of short sections was counted as they were produced and therefore the number of short vectors required to produce a normalised length vector may be calculated. The sum of the relative angles and a proportion of any intermediate vectors composing a normalised length vector is calculated. The resulting set of vectors describe the boundary of the object and do not include any position, orientation or size information.

The number of vectors used for the normalised contour was chosen to be 128. Appendix C shows the effects of chosing various numbers of vectors, the choice of 128 vectors was made since there seemed to be little appreciable improvement in the images by using more than this number.

### 4.5.4 Difficulty of normalisation

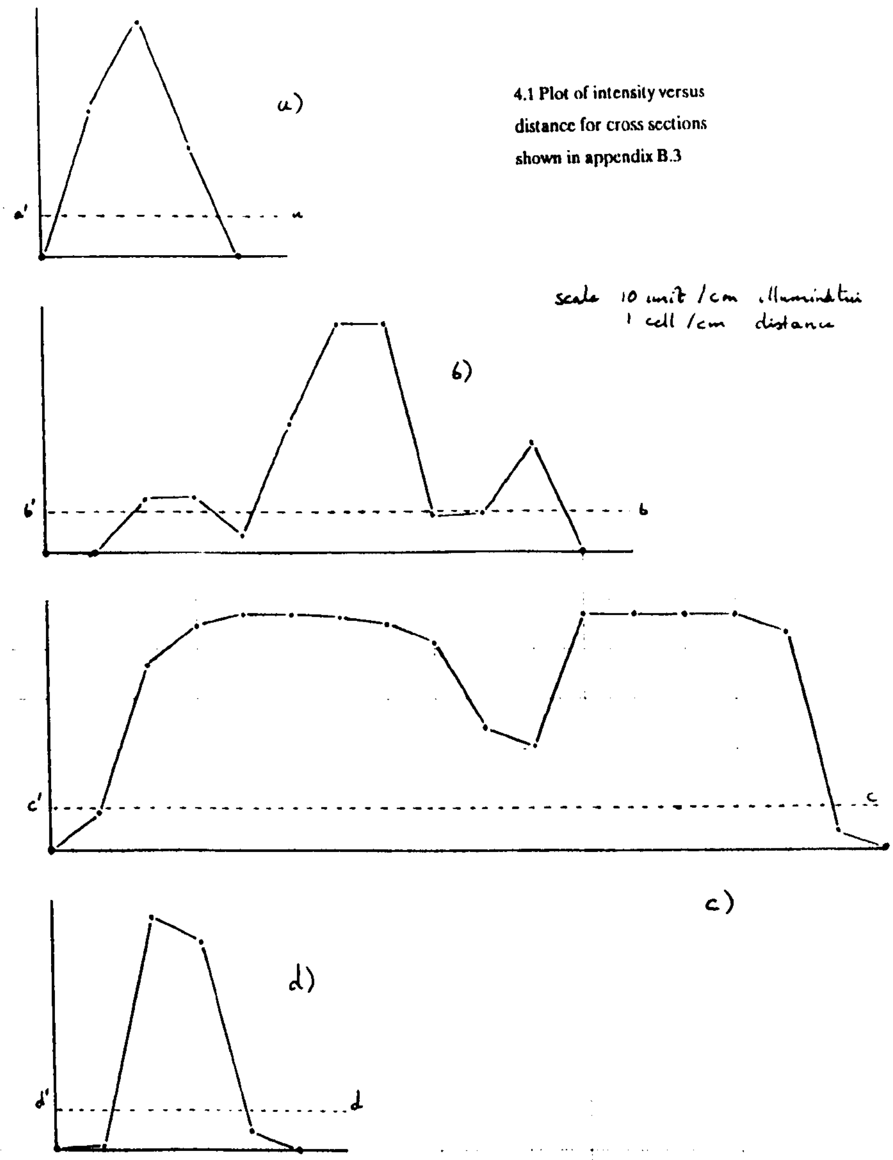
This method produces an approximation to the object contour, however it should be noted that the results of this algorithm have included an inherent error. It can be seen from figure 5.2 that the ends of the contour produced by this method do not meet up. The cause of this arises from the cumulative effect of the discrepancy in length between the line drawn through the points defining the start and end of groups of a short section lines, and the length of the vectors. The sum of lengths of groups of short sections is equal to the length of a normalised vector. However the distance between end points of the group will clearly be shorter if there are a number of sharp turns in the group than if the group is part of a straight line. There is usually only a single turn within any group since the number of output vectors is of the same order of magnitude as the number of original input vectors, and the effect tends to cancel out over a closed contour as any extension when moving away from the contour start is cancelled by an extension in moving back to close the contour.

The underlying cause of the problem comes from assuming that the overall length of the contour remains constant when the number of vectors comprising the contour are changed, even if the product of [number of vectors] * [length of vector] is maintained constant. This assumption makes calculation of a normalised vector set simple and allows for the calculation of the set at one pass through the short vectors and is most useful in this respect.

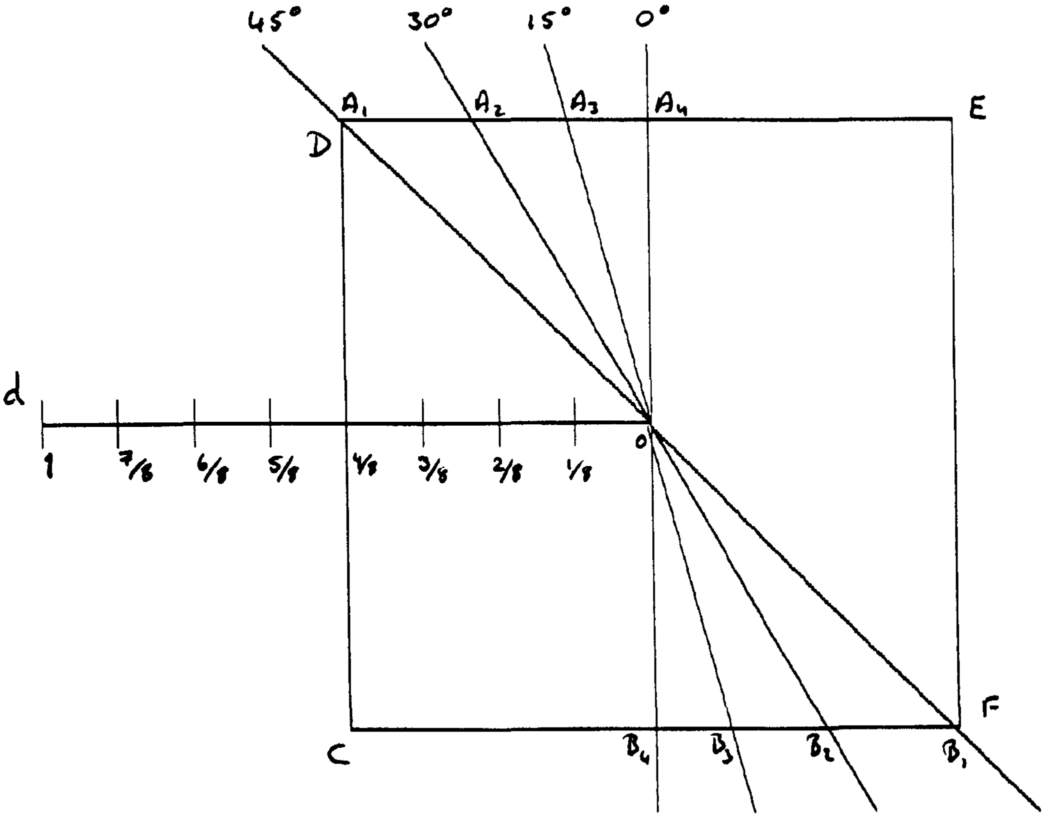
A solution to the problem can be found using an iterative method to determine the correct length of the normalised vectors. By estimating the length of the required vectors a contour set may be constructed by stepping round the contour as shown in figure 4.5. The advantage of this method is that each turning point found defining the vectors must be on the contour, and after a small number of repeated attempts it is expected that the normalised vector set will by very close to defining the shape of the original contour with small error.

The disadvantages of the approach are firstly that it requires several passes through the contour, where for each short contour a sine and cosine calculation are required. This in turn means that the points must be calculated using more precise mathematics than hitherto. Secondly it can be shown that the method will not produce a result unconditionally. Consider the diagram in figure 4.6. It can be seen that if the contour starts at point A and has estimated length b then the next step in the contour will start at point B. If this length were found to require too many steps then the step length would be increased, perhaps to length c. This would then give a step point at C and thus step over many more of the short vectors than in the previous iteration thus requiring the vector to be shortened. It is conceivable that the method would produce a situation that has no possible solution. However, the error involved will be less than one vector length, and will therefore generally be much smaller than given by the method currently used.

## 4.6 Diagrams



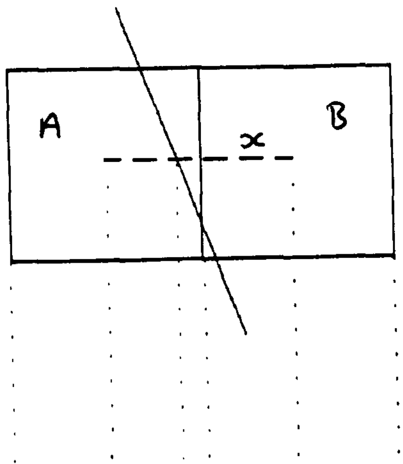

4.1 Plot of intensity versus distance for cross sections shown in appendix B.3



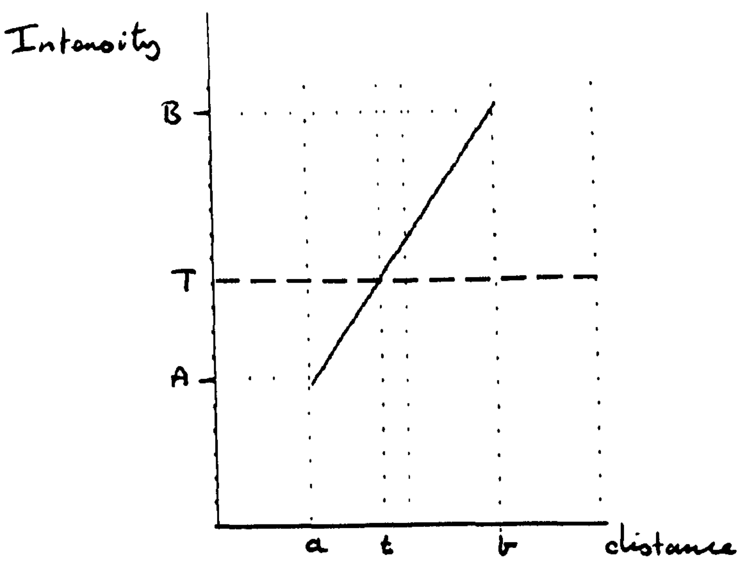

d

| | % | 0 | 1/8 | 2/8 | 3/8 | 4/8 | 5/8 | 6/8 | 7/8 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0° | 50 | 37 | 25 | 12 | 0 | 0 | 0 | 0 | 0 |
| x | 15° | 50 | 37 | 25 | 12 | 3 | 0 | 0 | 0 | 0 |
| | 30° | 50 | 37 | 25 | 14 | 7 | 2 | 0 | 0 | 0 |
| | 45° | 50 | 38 | 28 | 20 | 12 | 7 | 3 | 1 | 0 |

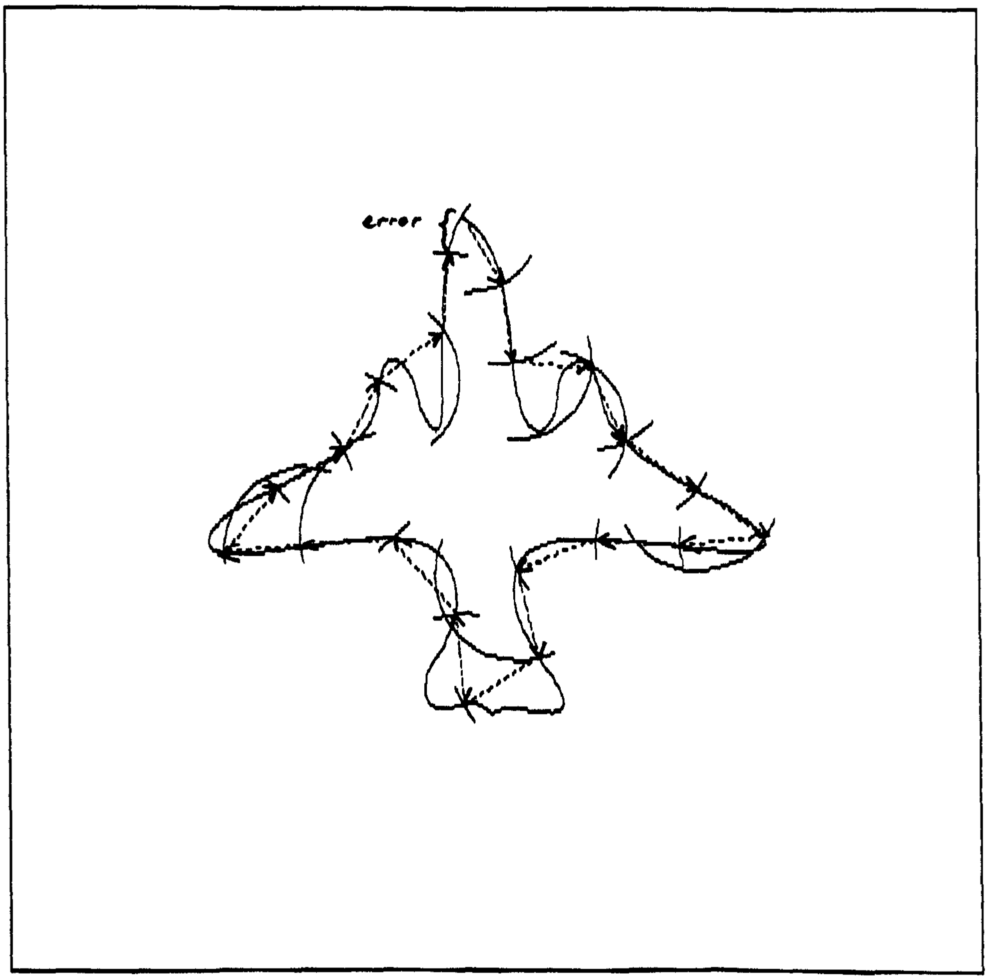
4.2 Region bounded by $A_x$ $B_x$ C D as a percentage of CDEF



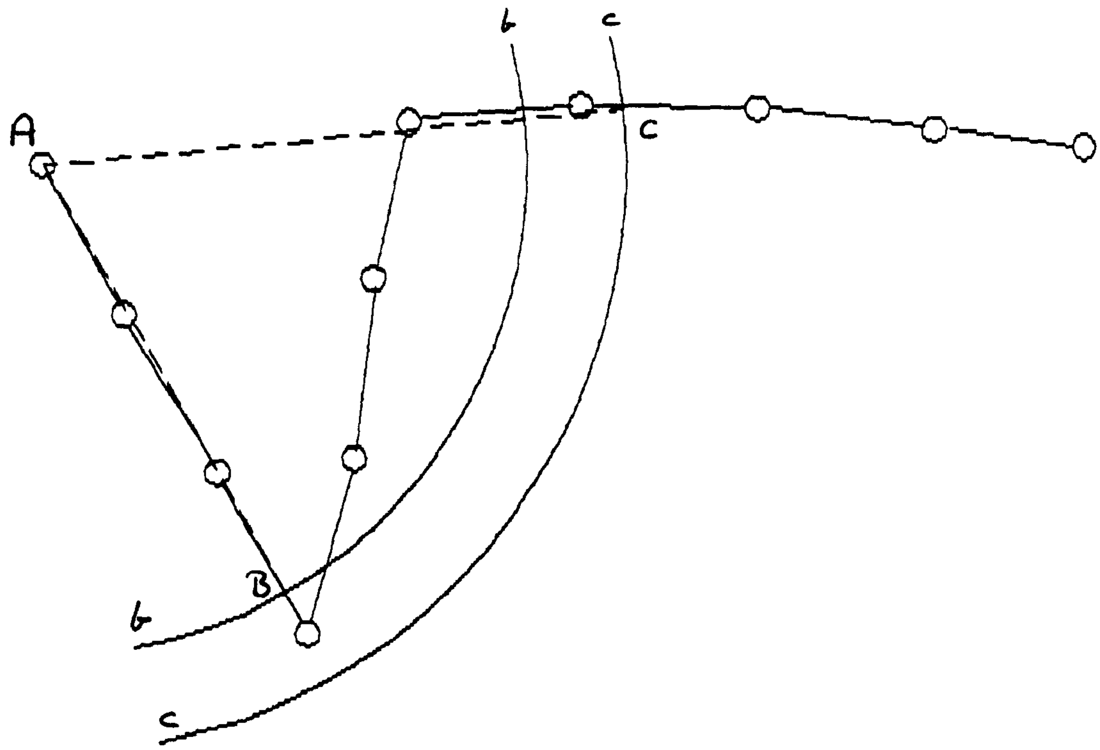

4.3 Adjacent cells




4.4 Assumed intensity variation between two cells




4.5 Iterative contour tracing

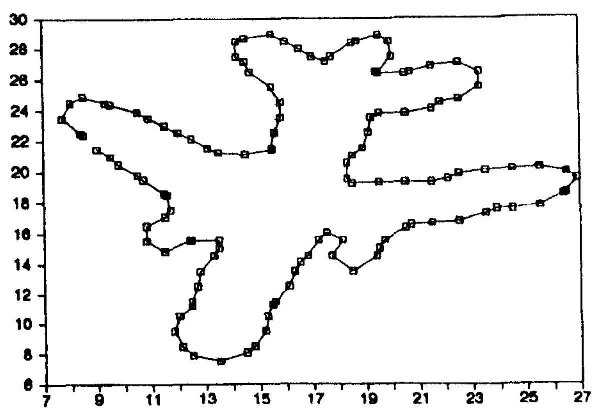



4.6 Pitfall in the iterative solution to normalised contour matching

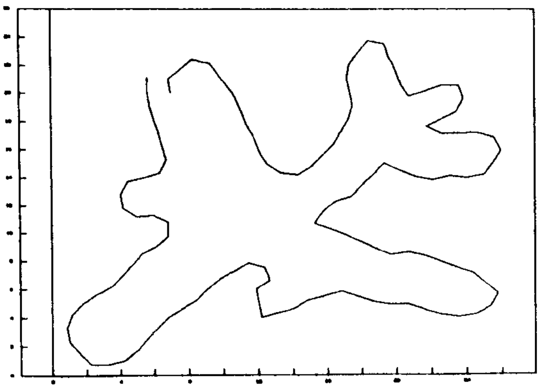
# Recognition Processes

## 5.1 Introduction

A length normalised contour is obtained for an image as described in the chapter 4. Figure 5.1 illustrates a typical example of the contour around a test image, and figure 5.2 shows the normalised contour that has been obtained from this pattern.




5.1 Original image 90015, showing sub-pixcel data points



5.2 Contour extracted for 90015

Once a normalised description has been obtained it must be compared with a library of patterns to determine the type of the described pattern and ascribe some label to it. A number of points arise from this problem.

*i)* The number of patterns required to compose a complete library of patterns is unknown.

*ii)* The resolution of details required to adequately define a pattern is unknown.

*iii)* The orientation of a test pattern relative to any member of the library of patterns is undefined.

*iv)* The required degree of correlation of test pattern and pattern library member to define a match or non-match is unknown.

*v)* The best method of comparison must be found.

When all of these problems have been resolved it will be possible to recognise simple patterns within the set defined by the assumptions in chapter 4.

## 5.2 Correlation Methods

### 5.2.1 Correlation by magnitudes of differences

The first method used to determine the correspondence between two vector descriptions $c_t$ and $c_l$ was to take the sum of the magnitudes of differences between corresponding vectors on the contours. This gives a resulting value F where

$$F = \sum_{i=0}^{N-1} |c_l[i] - c_t[i]|$$

For all examples used N in the above formula is 128 and $c_x[i]$ is the value of the $i^{th}$ element of the contour x. This matching process takes no account of the starting point from which contour extraction takes place, thus a modified form was used for matching contours.

$$F = \min_{i=0}^{N-1} (f[i])$$

where

$$f[i] \qquad = \sum_{j=0}^{N-1} |c_0[j] - c_1[(i + j) \bmod N])|$$

In this case the test contour is compared with the reference contour for each of the 128 possible contour starting points. It should be noted that no attempt has been made to remove the mean value at this stage. It is known from the algorithm used to extract the contours that all contours turn through exactly $360^{\circ}$, thus the mean turn for every contour description is 360/N, which is $2.81^{\circ}$. As the mean is the same for all contours it is removed by the correlating algorithm, thus no other processing is required.

The function as described has the advantage that it is comparatively fast since it uses integer arithmetic at all stages, and only involves mathematics which can be performed wholly within the CPU of all popular micro-computers. It is also complete, in that it obtains the best correlation result, regardless of the starting point of contours. It is therefore supposed that for many applications this method would be the most appropriate.

The drawbacks with this method are that it is comparatively insensitive to significant deviations of a small number of vectors, and that the value of the correlation function output is not set within any fixed scale.

A solution to the problem of scaling would be to divide the result by the correlation of the reference contour with the contour defined by $f[j] = {}^{2\pi}\!/_{N}$. This would then normalise the correlation function to give a value 1 when correlated against a circle. If the test contour were circular the problem of inaccuracy would arise, since a small change in the contour difference would produce a large change in the correlation function output, however in practical images this problem will rarely arise.

### 5.2.2 Correlation coefficient

The second correlation function used is the more usual function defined by

$$f'[j] = \frac{\sum_{i=0}^{N-1} C'_t[i] \times c'_l[k]}{\left( \sum_{i=0}^{N-1} C'_l[i]^2 \times \sum_{i=0}^{N-1} c'_t[k]^2 \right)^{1/2}}$$

where $k = (i+j) \bmod N$

and $F = \max(f'[0], f'[1] \ldots f'[N-1])$

The function $c'_x[]$ is defined as being the function $c_x[]$ with the mean value of all $c_x[]$ subtracted. Subtraction of the mean value is not strictly necessary since it is the same for all contours, being a total of one full circle over the sum of all vectors. Removing the mean value does, however, improve the discrimination of the function. It should be noted that this function differs from the previous function in that it gives a value of similarity in the range 1 to -1, not the value for difference.

A graph of the correlation coefficients $f'[0]$ to $f'[N-1]$ (i.e. for each correlation starting point) for 90/015 ($c_t[]$) correlated with 90/090 ($c_l[]$)is given in figure 5.12. The first differential of this graph is given in figure 5.13. This shows that at the point of best match the slope of the first difference is large, thus accurate registration of the two patterns is possible.

### 5.2.3 Application

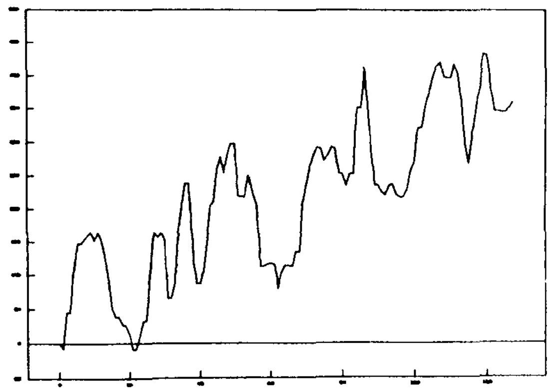
The correlation function given above was applied to a range of contour images, using the image 00/090 as the reference contour. The values of the peak correlation (F), and a value the position at which the peak occured (d) were recorded. The value d was used to produce a set of normalised images whereby the start point was the same for all patterns within the set. Similar tests were performed using 00/000 and 00/090 as references for a sub-set of patterns, and the resulting contours were used for subsequent tests.

## 5.3 Integration

### 5.3.1 First integral

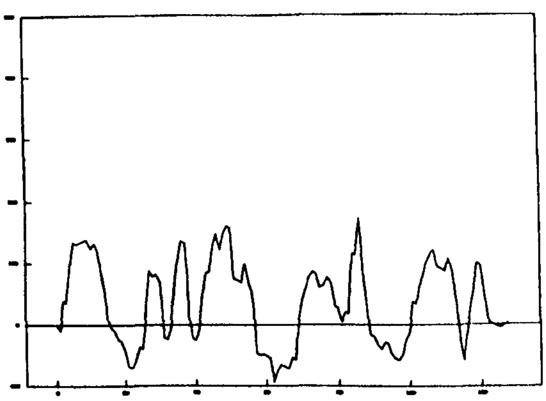
One problem anticipated in the contour length normalisation procedure is that the overall contour tends to follow a set of absolute vectors. This in turn means that if the contour deviates from its defined position at one step, an equal and opposite deviation will be imposed in the next step. This will be in addition to any error introduced in this further step.

A solution to this problem can be found by integrating the relative vector set to produce a normalised absolute vector set. In the process of following a closed curve a complete circle will be described, imposing a ramp function upon the output of the absolute vectors, as shown in figure 5.3.



5.3 Bearing around contour

This in turn re-introduces arbitrary orientation information to the contour description. Removing this ramp function by subtracting 2/N from each element of the contour description before integration will remove this absolute orientation information (figure 5.4). The resulting description can therefore have its start position moved simply by redefining which element of the description is considered to be the first element of the contour.



5.4 Bearings around contour with circle removed

The constant of integration introduced by the above step approximates to the distance by which the start position of the contour deviates from a normalised contour description of a perfect circle. It depends only upon the shape of the curve at the start position, and thus imparts no additional information. For this reason it is removed.

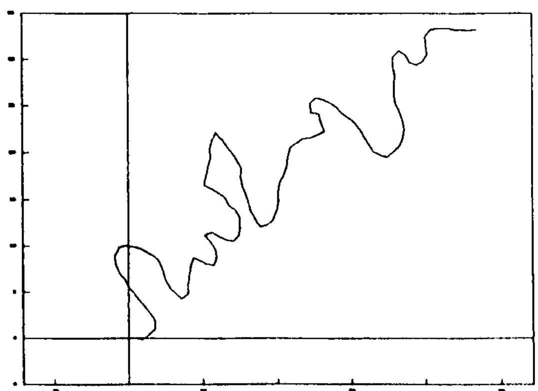
The output of the algorithm described is therefore suitable for arbitrary rotation and simple correlation in the same way as the relative vector set, but has the advantage that cumulative errors are minimised.

### 5.3.2 Second integral

The first integral of the relative vector set corresponds to the absolute bearing of each item of data, in the same way the second integral approimately corresponds to the position of the end points of vectors. It has been argued that much of the information in an image is associated with the shape of objects within the scene, and it is this shape information that has been extracted by the algorithm described so far. It was also felt by the author that certain features of an object may carry more information than others, for example the intricate turning pattern of the tailplane of the aircraft may carry sufficient information for recognition, whereas the straight leading edge of a wing would not, even though both sections might have the same number of data elements. In order to assist with the assessment of this the second integral of the relative vector set was used. Whilst the second integral is not exact shape information, features can be recognised from plots of it against number of data elements. True position information is obtained from

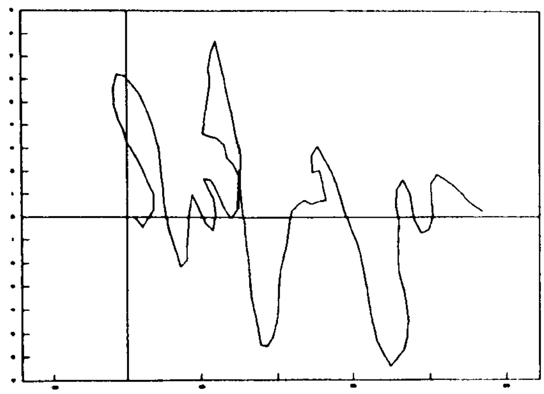
$\int e^{i\theta} d\theta$,

that is $\int(\cos\theta + i\sin\theta)d\theta$, whereas we have used $\int \theta d\theta$. It can be seen that if all $\theta$ is small, and therefore $\int \cos\theta d\theta$ is approximately 0, then the approximation of $\int \theta d\theta$ to $\int e^{i\theta} d\theta$ is good. Figure 5.5 shows the graph of the second integral using the complex method.

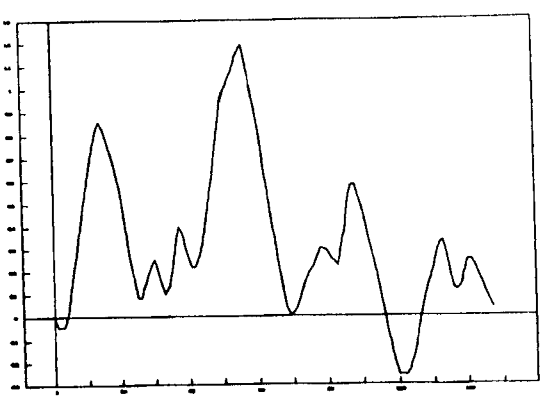


5.5 Integrated contour using complex representation

In practice the integral of is made to be zero by removing the ramp function associated with turning a full circle. The value of may be large in a number of individual instances around the contour, thus the first condition above does not hold. The graphs in figure 5.6 and 5.7 show the difference between the two types of plot. Although there are significant differences between the plots it can be seen that the principle features are clearly visible in both plots, thus it is proposed that the simple calculation of $\int \theta d\theta$ should be used.



5.6 Complex contour with constant of integration removed

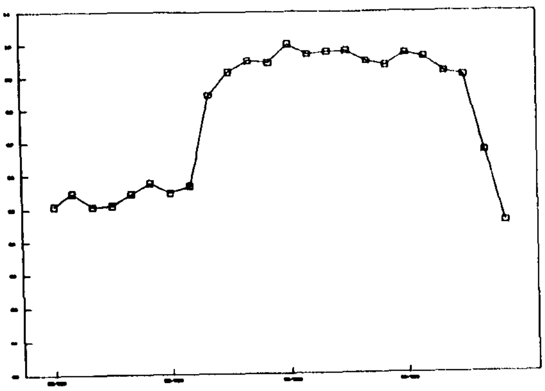


5.7 Integral of contour using real components only

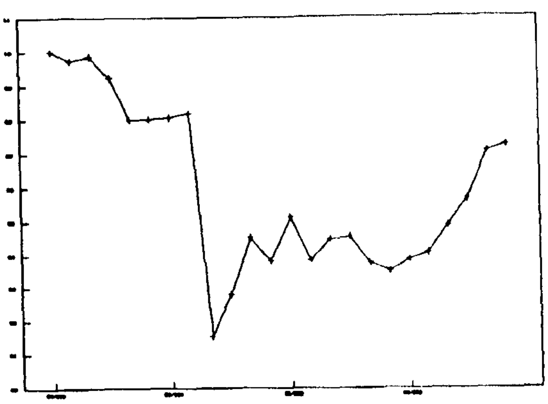
This also has the advantages that the simple integral is a single valued function whereas the exponential function gives a complex result, and that the high frequency component of the contour is attenuated reducing the overall noise. In the choice of simple integral, exponential integral or no integration at all, all approaches have merits, and as there is a reversible transform to obtain each from either of the others, there is no information lost. It is believed that simple integration is most suitable for research purposes, but that this step may be omitted once analysis and comparison techniques are better understood.

## 5.4 Correlation

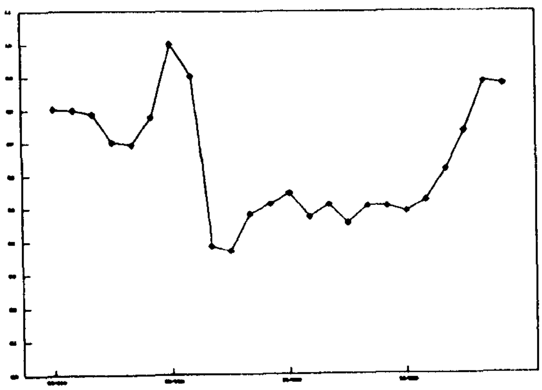
Three test patterns were chosen from the sample set as references against which all patterns within the sample were correlated. The sample set is shown in figure 5.14, with the reference patterns highlighted by a label. Each pattern was correlated against each reference in each of the 128 possible positions of start point. The peak value of each of these tests is given in the table in figure 5.15. The graph in figure 5.8 shows the values of correlation against the reference 90/090. It can be seen that all of the aeroplanes seen from above, and the 6 in the set ee/nnn, where ee is 00 or 90 and nnn is any of 075, 060 or 045, have correlation coefficients of greater than 0.9. This is illustrated in figure 5.11 by the line labelled (1), where all points above the line represent the patterns which have a corrlelation coefficient of greater than 0.9. In a similar manner the test patterns closeley matching the library patterns (2) 00/000 (figure 5.9) and (3) 00/090 (figure 5.10), are also illustrated on this diagram.



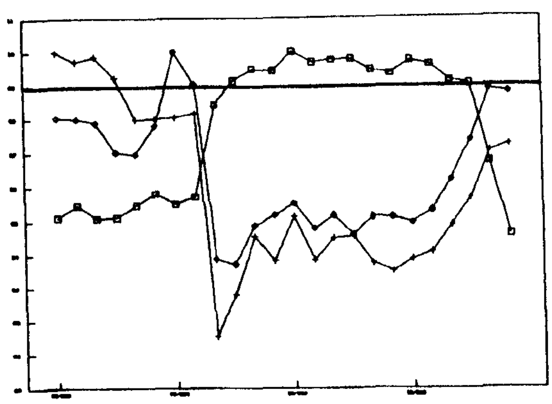
5.8 Maximum correlation of sample set with 90/090



5.9 Maximum correlation of sample set with 00/000



5.10 Maximum correlation of sample set with 00/090




5.11 Maximum correlation of sample set with all references

The mean correlation of all patterns in the range 90/nnn with 90/090 is 0.973, having a standard deviation of 0.022. This gives an indication of the amount of noise that has been introduced by the digitiser and the algorithms.

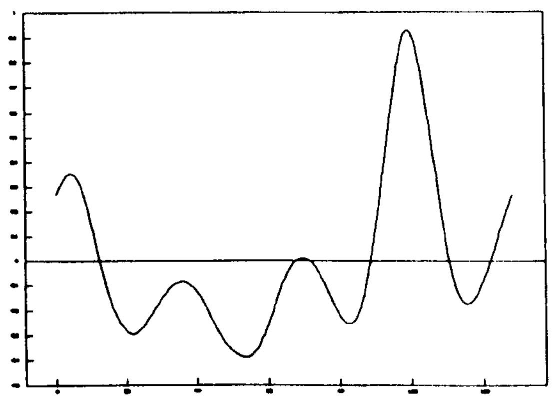
By allowing a correlation of greater than 0.9 to identify a test pattern with a reference pattern we have included 19 of the 24 views of the object into just 3 sets. With the segmentation technique used at least 3 more sets will be required to describe the remaining 5 objects, however the objects in 00/060 and 00/075 have been significantly altered by the presence of wheel studs as detailed in chapter 4, and it may be that an improved segmentation technique would include these in adjacent sets.

The mean and standard deviation of the test object correlations with their respective reference objects are:-

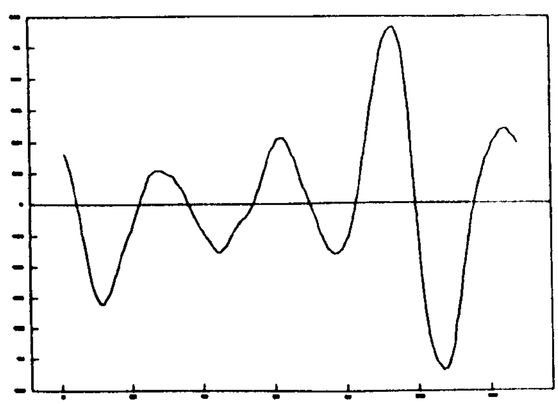
| reference | set | mean | standard deviation |
|---|---|---|---|
| 90/090 | 1 | 0.967 | 0.022 |
| 00/000 | 2 | 0.970 | 0.034 |
| 00/090 | 3 | 0.952 | 0.0690 |

From this it can be seen that the mean correlation is generally much greater than the 0.900 used as a threshold. For example the resolution of the reference set could be increased by using a threshold correlation value of 0.92. This would increase the discrimination of the system at the expense of increasing the number of tests required to find the correct match, and increasing the number of library samples. It is clear from the mean correlation values and standard deviations from these means that noise associated with image processing would not unduly influence the discrimination of the system using this threshold.

It is supposed that improving the digitiser hardware by increasing accuracy of sampling and by using many more cells the amount of noise introduced will be reduced. The calculation of the normalised contour will take proportionately longer, however since the contour is normalised to a fixed number of elements, any processing after this will be unaffected. It has also been noted that the method of scaling the length of the contour is far from ideal, and this may also be a source of noise which could be improved upon.



5.12 Correlation coefficient versus relative rotation



5.13 Rate of change of correlation coefficient

Figure 5.12 shows a plot of the correlation values obtained by relating the pattern 90/015 to the reference pattern 90/090. There are three starting positions that produce correlation coefficients very close to the peak value. This suggests that the number of vectors used to describe this particular object is greater than required, and that as few as one third as many (43) may be sufficient. The effect of such a reduction may be sufficient to cause some small features on the aeroplane to be lost (e.g. the wing engines). A halving of the number of vectors may, however, be acceptable, and would cause an increase in speed of correlation by a factor of 8. Since the correlation is the most time consuming of any of the processing sections, this benefit would be most desirable.

Supporting evidence for this is given by the differential of the contour coefficients with respect to start position (figure 5.13). It can be seen that the graph has a very well defined and linear differential over the region near the peak, and this in turn indicates that a smaller number of points will serve to define the peak location and height.

### 5.4.1 Correlation Alignment

The most time consuming part of the algorithm described is finding the peak correlation. This occurs because it is necessary to correlate all of the potential test contour starting positions with the reference contour. There is no possible absolute starting position for a contour, firstly because such a concept is meaningless in most cases, and secondly because for any arbitrarily defined start position to be used the object in question must first be identified.

### 5.4.2 Fourier transforms for correlation guidance

Tests were made which could potentially provide a short cut using Fourier Transforms. The contour descriptions are single valued continuous sets, and are thus most amenable to Fourier Transformation. The physical meaning of this transformation is difficult to describe, however, the $0^{th}$ harmonic will contain the information associated with the loop nature of contours, and thus for the first integral of the relative vector set it will always have the value 2.81 (360/128). The first harmonic is related to the position of the centre of gravity of the pattern, and indicates any bias in the density, and hence complexity of vectors on one side of the pattern with respect to the other side.

The second harmonic contains information about the length versus breadth of an object, and it is this information that may improve the speed of the correlation process. If the object is longer in one dimension than the other then the magnitude of the second harmonic phasor will be non zero. The phase of the second harmonic vector will determine the orientation of the object, and thus by causing all suitable reference patterns to have contour start positions synchronised with the phase of the second harmonic, any corresponding test images need only be checked against a limited number of reference contour start positions.

The table in Appn. D shows the value of the phase of the second harmonic for all patterns in the set 90/nnn (i.e. seen from above). The vector set start positions have been normalised with respect to 90/090 using the correlation coefficient method already described, thus any value in the second harmonic represents the error in using this method with respect to the correlation coefficient method. The correlation coefficient method is known to be accurate to within one position. It can be seen that whilst the mean value of error is small the standard deviation is large-approximately 4 positions, thus it is clear that this method is not nearly as accurate as the correlation coefficient method. However, in the worst case the Fourier method is 31 degrees in error, this represents 10 vectors, thus 10 vectors to each side of the two possible indicated orientations must be searched. This gives 40 correlation coefficients that must be calculated instead of 128 which would be a worthwhile saving. It is supposed that the magnitude of the second harmonic phasor will give some indication as to the reliability of the phase information, however this has not been tested and would be a most interesting topic for further investigation.

## 5.5 Fourier Correlation

### 5.5.1 Phase Angle Differences

A further set of tests were performed to discover if the Fourier Transform of the doubly integrated vector sets could itself be used for correlation, or give an indication of the scale of features that were correlating.

The correlation coefficient is given by the cosine of the difference in phase angle for a pair of Fourier components. The results given in table 5.16 show that for similar images the correlation coefficients are close to 1 for harmonics as high as the $20^{th}$. Instances in a correlated pair of patterns where the correlation coefficient is very low have been shown to be due to the magnitude of the phasor being small, and thus far more susceptible to noise.

It can be seen that the correlation coefficient is always greater than 0.4 up to the $8^{th}$ harmonic, and that cases of matching harmonics up to the $20^{th}$ or above can be found. The implications of this are that the Fourier Transform of the vector sets may supply a more robust measure of correlation. Also the most appropriate filter for noise reduction will be given by the phasor magnitude for each harmonic of a reference pattern. This allows for automatic filtering, and supplies a measure of the reliability of the correlation. A further implication is that by using a number of harmonics, possibly with weightings supplied by the magnitude of the phasors, it is a comparatively simple matter to calculate the relative orientation of a test object with respect to a reference object.

This can be done by dividing the difference between the test phase angle and reference phase angle by the harmonic number for each harmonic. The mean of the values given represents the mean phase difference between the two patterns, and this is directly proportional to relative orientation.

### 5.5.2 Minimising Phase Angles

An alternative strategy would be to minimise the mean value of phase angle for reference and test patterns. The logic for this closely follows that for normalisation using only the second harmonic as described in the previous subsection, however it is felt that by appropriate weighting of results using the magnitude information and low pass (low harmonic only) filtering this method will yeild more accurate results. Also it has been noted that correlation is only good up to 20 harmonics, thus only 20 data values need to be correlated for optimal results. Part of the reason for this is found in the nature of the noise introduced to the contour.

**Types of Noise**

The first type of noise is that described previously as due to an error carried from one bearing to the next. Although the integration process will reduce this, there will nevertheless be some ripple at the contour step scale, and this will be found in harmonics greater than the $70^{th.}$

The second type of noise is due to the interpolation. Noise introduced by the camera that increases the value of a cell or pair of adjacent cells will cause a bump in an otherwise straight line. This will appear as noise in harmonics lower than the $30^{th}$ if two or three cells in a row have similar noise.

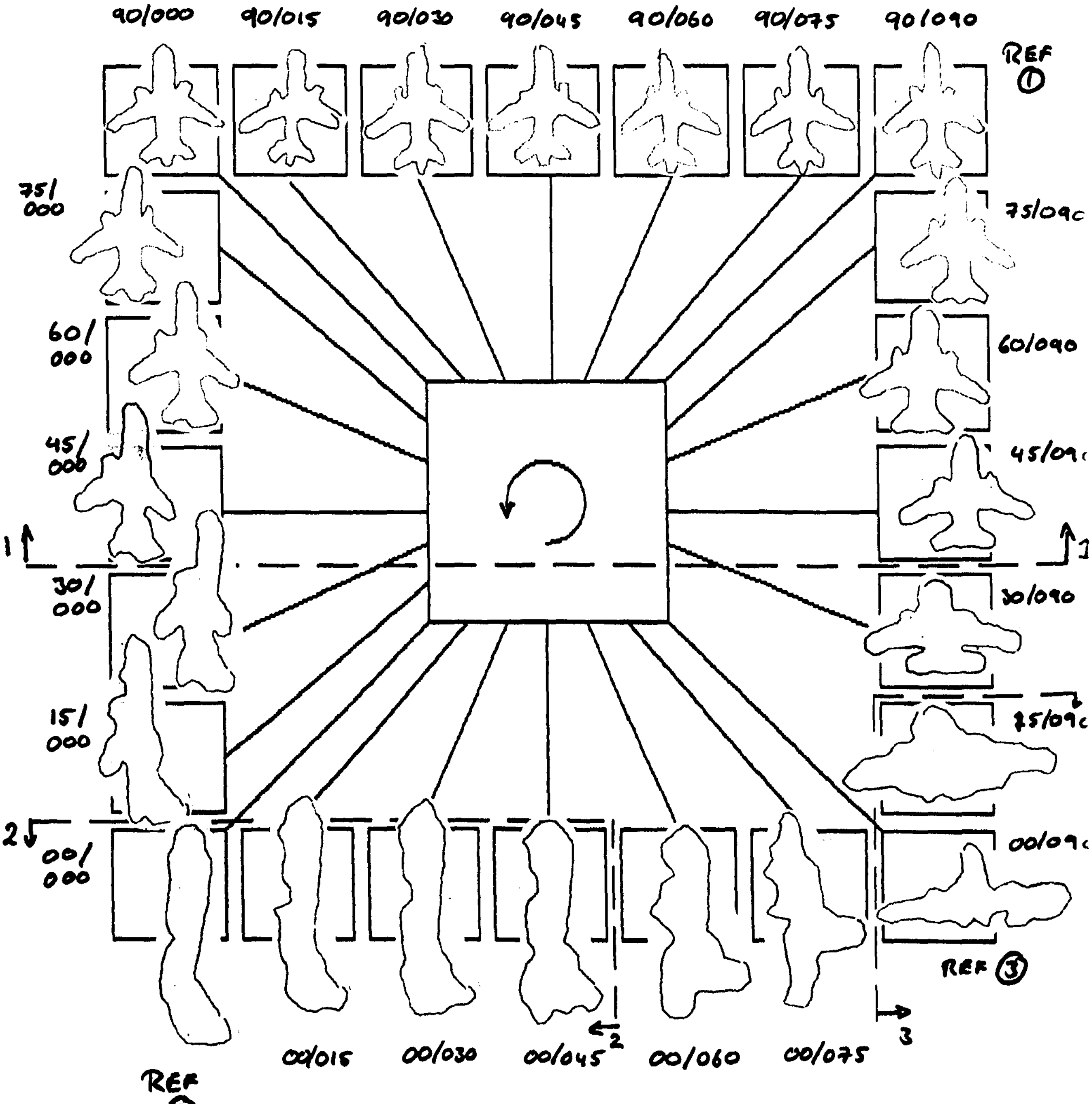
### 5.5.2.1 Direct Correlation of Normalised Phase Angles

Given that the above is correct then the resulting description will not only have lost orientation information, it will have a normalised orientation imposed such that matching of test against reference can be performed by direct comparison without recourse to iteration, or other methods of repeated correlation. The saving in processor time is very likely to make the lengthy transformation most worthwhile. It should be noted that hardware Fourier Transforms are available which make this process very fast indeed.

### 5.5.3 Results of data reduction

An estimate has been made of the number of reference contours that would be needed for a complete set that matches any test contour with a correlation coefficient of greater than 0.9. The estimated number is 29, thus requiring 5 bits to represent this information. An additional 5 bits of information can be obtained from the recorded orientation by extraction from the direction of the first vector of the test contour compared with the reference contour. The positional information is not regarded as useful in identifying the pattern though it is available as a by product of searching for the contour start position.

From this it can be seen that 10 bits of information have been derived representing the presence or absence of one of 29 shapes and their orientation from the 8192 bits originally digitised. It is also noted that if any of the 29 shapes is registered then the single bit of information denoting "aeroplane" is true.




5.14 Test object views used for correlation experiments

| Object | Thr | Vectors | Reference Object 90/090 | 00/000 | 00/090 |
|---|---|---|---|---|---|
| 00/000 | 28 | 339 | 0.510 | *1.000 | 0.804 |
| 00/015 | 24 | 356 | 0.546 | *0.972 | 0.800 |
| 00/030 | 22 | 337 | 0.507 | *0.986 | 0.788 |
| 00/045 | 21 | 325 | 0.510 | *0.922 | 0.701 |
| 00/060 | 21 | 299 | 0.545 | 0.798 | 0.694 |
| 00/075 | 17 | 318 | 0.578 | 0.801 | 0.780 |
| 00/090 | 13 | 358 | 0.548 | 0.804 | *1.000 |
| 15/090 | 20 | 331 | 0.568 | 0.817 | *0.903 |
| 30/090 | 25 | 460 | 0.844 | 0.154 | 0.387 |
| 45/090 | 44 | 518 | X0.913 | 0.280 | 0.372 |
| 60/090 | 45 | 576 | X0.948 | 0.452 | 0.484 |
| 75/090 | 48 | 672 | X0.942 | 0.380 | 0.515 |
| 90/090 | 45 | 704 | *1.000 | 0.510 | 0.548 |
| 90/075 | 42 | 730 | *0.968 | 0.383 | 0.475 |
| 90/060 | 41 | 734 | *0.975 | 0.446 | 0.513 |
| 90/045 | 43 | 705 | *0.978 | 0.452 | 0.456 |
| 90/030 | 42 | 713 | *0.945 | 0.374 | 0.509 |
| 90/015 | 42 | 727 | *0.933 | 0.350 | 0.507 |
| 90/000 | 44 | 696 | *0.972 | 0.385 | 0.491 |
| 75/000 | 45 | 668 | X0.960 | 0.406 | 0.526 |
| 60/000 | 43 | 572 | X0.912 | 0.484 | 0.617 |
| 45/000 | 39 | 514 | X0.901 | 0.561 | 0.733 |
| 30/000 | 27 | 461 | 0.669 | 0.702 | 0.884 |
| 15/000 | 29 | 428 | 0.455 | 0.720 | 0.876 |

| Reference | Mean | Std. Dev | |
|---|---|---|---|
| 90/090 * | 0.967 | 0.022 | 90/nnn only |
| 90/090 X | 0.950 | 0.029 | > 0.9 |
| 00/000 | 0.970 | 0.034 | > 0.9 |
| 00/090 | 0.952 | 0.069 | > 0.9 |

Thr Threshold value used

Vectors Number of vectors in coarse contour

5.15 Correlation of reference objects with the test set of objects

| 90/000 | 90/015 | 90/030 | 90/045 | 90/060 | 90/075 |
|---|---|---|---|---|---|
| 0.999 | 0.999 | 0.999 | 0.999 | 0.999 | 0.999 |
| 0.990 | 0.911 | 0.991 | 0.996 | 0.961 | 0.995 |
| 0.998 | 0.982 | 0.998 | 0.995 | 0.998 | 0.998 |
| 0.994 | 0.937 | 0.945 | 0.977 | 0.992 | 0.997 |
| 0.999 | 0.993 | 0.985 | 0.997 | 0.998 | 0.999 |
| 0.999 | 0.994 | 0.983 | 0.987 | 0.999 | 0.999 |
| 0.974 | 0.935 | 0.815 | 0.943 | 0.968 | 0.999 |
| 0.995 | 0.998 | 0.984 | 0.989 | 0.996 | 0.999 |
| 0.964 | 0.959 | 0.975 | 0.999 | 0.991 | 0.998 |
| -0.206 | 0.408 | 0.467 | 0.681 | -0.032 | 0.996 |
| 0.694 | 0.924 | 0.800 | -0.941 | 0.972 | 0.750 |
| 0.999 | 0.967 | 0.937 | 0.990 | 0.949 | 0.999 |
| 0.987 | 0.124 | 0.290 | 0.999 | 0.612 | 0.968 |
| 0.998 | 0.724 | 0.936 | 0.797 | 0.935 | 0.719 |
| 0.999 | 0.913 | 0.285 | 0.924 | 0.947 | 0.984 |
| 0.999 | 0.971 | 0.998 | -0.232 | 0.990 | 0.994 |
| 0.991 | 0.956 | 0.848 | 0.997 | 0.979 | 0.985 |
| 0.018 | 0.901 | 0.967 | 0.998 | 0.999 | -0.620 |
| 0.968 | 0.980 | 0.695 | -0.651 | 0.504 | 0.999 |
| 0.973 | 0.996 | 0.529 | 0.895 | 0.854 | 0.896 |

**5.16 First 20 harmonic correlations of vertical set matched against 90/090**

# Conclusions

Research during the first year on automatic pattern recognition concentrated upon transform techniques. Much of the time was spent on Fourier Transforms and several potential image location methods have been studied.

These are:-

*i)* Target location by averaging the gradient of the phase plane to detect phase plane tilt.

*ii)* Target location by Fourier Transformation of phase plane information, to detect the phase plane tilt.

*iii)* Comparison of Walsh-Hadamard and Fourier Transforms.

*iv)* Target location by edge detection using Walsh-Hadamard Transforms as filters.

*v)* Target location by relating phase components of the Fourier Transform with the modulus components of a Fourier Transformed square.

In each case the major drawback has been thermal noise and background information which is included in the transform, and which tends to obscure the effect under study.

A great deal of empirical knowledge has been gained about the properties of images and transforms from the first year's work. However it appears that transform techniques do not provide the whole answer to image analysis problems. The greatest difficulty with this approach is that it is not clear what responses are useful, partly because the effects observed using artificial images have proved to be considerably less pronounced when swamped by noise found on a real thermal image, and partly because the usefulness of any particular property of a transform is not evaluated quantitatively by this approach.

Transform techniques probably have great value in image analysis, but they cannot be considered in isolation. The need for a transform to reduce redundant information in an image may be dictated by a system, or the use of a transform may improve an existing system. However in each case it is the system for classifying images which must be produced first, and an appropriate transform may then by used to enhance its operation.

The use of contour description has by contrast been very successful. Reliable and accurate descriptions of complex features have been obtained using the contour following algorithm described, and a great deal of promise is shown by the use of Fourier Transforms as the final step in the normalising procedure.

The use of fully visible objects may be seen by some as a severe drawback to the algorithm, since in many cases objects are partially occluded in scenes by other objects. However, it is felt that by producing the normalised description of a contour the groundwork for a semantic pattern recognition process has been established. Consideration should be given to contour description of line segments rather than an entire object. The Fourier Transform of such descriptions will not necessarily match in phase angle directly, however a shift of the results will scale the effective length of the contour, thus the problems of matching are reduced to one of shift and direct correlation on up to 20 data points. This is not as onerous a task as matching the original image matrices.

The results, therefore, present a useful adjunct to pattern recognition techniques, and it is hoped that further research into syntactic methods of image understanding will benefit from this study.

# References

1. Pavlidis T., "Algorithms for Graphic and image processing", 'Computer Science Press', 1982

2. Ballard D.H. and Brown C.M.,"Computer Vision", Prentice-Hall Inc. 1982,

3. Rosenfeld A., "Picture processing by computer", Academic Press Inc. 1969,

4. Niblack W., "An Introduction to Digital Image Processing", Prentice-Hall Inc. 1986

5. De Preist D.J. and Wegman,"Statistical Image Processing and Graphics", Marcel Dekker Inc, 1986

6. Gonzalez R.C. and Wintz P.,"Digital Image Processing", 2nd edition, Addison Wesley Publishing Co., 1987

7. Brigham E.O., "The fast fourier transform", Prentice-Hall Inc. 1974

8. Elliott D.F. and Ramamohan Rao K.,"Fast Transforms - Algorithms, analyses, applications", Academic Press, 1982

9. Keren D., Peleg S. and Schweitzer L,"Improving image resolution using sub-pixel motion", 'Pattern Recognition letters 5', pp223-226, 1987

10. Chanda B., Chandhuri B.B., Dutta Majumder D., "A Modified Scheme for Segmenting the Noisy Images", IEEE Trans. on Systems, Man and Cybernetics.

11. Keren D., Peleg S., Brada R., "Image Sequence Enhancements using Sub-pixcel Displacements", IEEE ref CH2605-4/88/0000/0742, 1988.

12. Lee Jong-Sen, "Statistical Image Processing and Graphics", Marcel Dekker Inc, ISBN 0-8247-7600-3, 1986.

FT.LOG:1 4-JAN-1984 11:21 Page 2

**** INPUT DATA ****

INPUT FILES REAL : SQFL1010 IMAGINARY : NULL0000

REAL PART LEFT HALF SCALE FACTOR = 10.0909

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 999 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**1. 8 by 8 rectangular test object**

FT.LOG;1 20-JAN-1984 10:10 Page 6
**** TRANSFORMED DATA ****
INPUT FILES REAL : [illegible] IMAGINARY : NULL0000
MODULUS LEFT HALF SCALE FACTOR = 510.2761

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 993 | 967 | 675 | 734 | 558 | [illegible] | 180 | [illegible] | 111 | 192 | 225 | 214 | 166 | 96 | 17 | 56 | 111 | 141 | 143 | 119 | 74 | 19 | 37 | 82 | 111 | 118 | 102 | 68 | 22 | 27 | 71 | 102 |
| 967 | 926 | [illegible] | 710 | 540 | 357 | [illegible] | [illegible] | 107 | 186 | 214 | 207 | 161 | 93 | 17 | 54 | 107 | 137 | 139 | 115 | 72 | 18 | 35 | 80 | 107 | 114 | 99 | 66 | 21 | 26 | 69 | 97 |
| 675 | [illegible] | 767 | 642 | 479 | 322 | [illegible] | [illegible] | 97 | 169 | 187 | 187 | 144 | 64 | 15 | 49 | 97 | 124 | 125 | 104 | 65 | 17 | 32 | 72 | 97 | 103 | 90 | 60 | 19 | 24 | 62 | [illegible] |
| 734 | 712 | [illegible] | 539 | 410 | 276 | 135 | [illegible] | 82 | 141 | 146 | 157 | 122 | 70 | 13 | 41 | 82 | 104 | 105 | 87 | 54 | 14 | 27 | 61 | 82 | 86 | 75 | 50 | 16 | 20 | 52 | 74 |
| 558 | 542 | [illegible] | 412 | 312 | [illegible] | 112 | [illegible] | 62 | 107 | 124 | 119 | 93 | 54 | 10 | 21 | 62 | 79 | 80 | 66 | 41 | 11 | 20 | 46 | 62 | 66 | 57 | 39 | 12 | 15 | 40 | 56 |
| 367 | [illegible] | [illegible] | 270 | [illegible] | [illegible] | [illegible] | [illegible] | 41 | 71 | [illegible] | [illegible] | 61 | [illegible] | 6 | 20 | 41 | 52 | 52 | 44 | 27 | 7 | 13 | 30 | 41 | 43 | 39 | 25 | 8 | 10 | 26 | 37 |
| 160 | 175 | 156 | 122 | 101 | 66 | 33 | [illegible] | 20 | [illegible] | 41 | 29 | [illegible] | 17 | 3 | 10 | 20 | 25 | 26 | 21 | 13 | 3 | 7 | 15 | 20 | 21 | 19 | 12 | 4 | 5 | 13 | 18 |
| 16 | 16 | 14 | 12 | 9 | 6 | 3 | [illegible] | [illegible] | [illegible] | 4 | 3 | 3 | 2 | 0 | 1 | 2 | 2 | 2 | 2 | 1 | 0 | 1 | 1 | 2 | 2 | 2 | 1 | 0 | 0 | 1 | 2 |
| 111 | 107 | 97 | 82 | 62 | 41 | 20 | 2 | 12 | 21 | 25 | 24 | 19 | 11 | 2 | 6 | 12 | 15 | 16 | 13 | 9 | 2 | 4 | 9 | 12 | 13 | 11 | 8 | 2 | 3 | 8 | 11 |
| 192 | 186 | 169 | 141 | 107 | 71 | 35 | 3 | 21 | 37 | 43 | 41 | 32 | 18 | 3 | 11 | 21 | 27 | 28 | 23 | 14 | 4 | 7 | 16 | 21 | 22 | 20 | 13 | 4 | 5 | 14 | 19 |
| 225 | 213 | 197 | 166 | 126 | 83 | 41 | 4 | 25 | 43 | 51 | 49 | 37 | 22 | 4 | 13 | 25 | 32 | 32 | 27 | 17 | 4 | 8 | 19 | 25 | 27 | 23 | 15 | 5 | 6 | 16 | 23 |
| 214 | 207 | 187 | 157 | 119 | 78 | 39 | 3 | 24 | 41 | 47 | 46 | 36 | 20 | 4 | 12 | 24 | 30 | 31 | 25 | 16 | 4 | 8 | 18 | 24 | 25 | 22 | 15 | 5 | 6 | 15 | 21 |
| 166 | 161 | 146 | 122 | 92 | 61 | 30 | 3 | 19 | 32 | 37 | 36 | 28 | 16 | 3 | 9 | 19 | 23 | 24 | 20 | 12 | 3 | 6 | 14 | 18 | 20 | 17 | 11 | 4 | 5 | 12 | 17 |
| 95 | 93 | 84 | 70 | 54 | 35 | 17 | 2 | 11 | 14 | 22 | 20 | 15 | 9 | 2 | 5 | 11 | 14 | 14 | 11 | 7 | 2 | 4 | 8 | 11 | 11 | 10 | 7 | 2 | 3 | 7 | 10 |
| 17 | 17 | 15 | 13 | 10 | 6 | 3 | 0 | 2 | 3 | 4 | 4 | 3 | 2 | 0 | 1 | 2 | 2 | 2 | 2 | 1 | 0 | 1 | 1 | 2 | 2 | 2 | 1 | 0 | 0 | 1 | 2 |
| 56 | 54 | 49 | 41 | 31 | 20 | 10 | 1 | 6 | 11 | 13 | 12 | 9 | 5 | 1 | 3 | 6 | 8 | 8 | 7 | 4 | 1 | 2 | 5 | 6 | 7 | 6 | 4 | 1 | 2 | 4 | 6 |
| 111 | 107 | 97 | 82 | 62 | 41 | 20 | 2 | 12 | 21 | 25 | 24 | 18 | 11 | 2 | 6 | 12 | 15 | 16 | 13 | 8 | 2 | 4 | 9 | 12 | 13 | 11 | 8 | 2 | 3 | 9 | 11 |
| 141 | 137 | 124 | 104 | 79 | 52 | 25 | 2 | 16 | 27 | 32 | 30 | 23 | 14 | 2 | 8 | 16 | 20 | 20 | 17 | 10 | 3 | 5 | 12 | 16 | 17 | 14 | 10 | 3 | 4 | 10 | 14 |
| 143 | 138 | 125 | 105 | 80 | 52 | 26 | 2 | 16 | 29 | 32 | 31 | 24 | 14 | 2 | 9 | 16 | 20 | 20 | 17 | 11 | 3 | 5 | 12 | 16 | 17 | 15 | 10 | 3 | 4 | 10 | 14 |
| 119 | 115 | 104 | 87 | 66 | 44 | 21 | 2 | 13 | 23 | 27 | 25 | 20 | 11 | 2 | 7 | 13 | 17 | 17 | 14 | 9 | 2 | 4 | 10 | 13 | 14 | 12 | 8 | 3 | 3 | 8 | 12 |
| 74 | 72 | 65 | 54 | 41 | 27 | 13 | 1 | 9 | 14 | 17 | 16 | 12 | 7 | 1 | 4 | 8 | 10 | 11 | 9 | 6 | 1 | 3 | 6 | 8 | 9 | 8 | 5 | 2 | 2 | 5 | 7 |
| 19 | 18 | 17 | 14 | 11 | 7 | 3 | 0 | 2 | 4 | 4 | 4 | 3 | 2 | 0 | 1 | 2 | 3 | 3 | 2 | 1 | 0 | 1 | 2 | 2 | 2 | 2 | 1 | 0 | 1 | 1 | 2 |
| 37 | 35 | 32 | 27 | 20 | 13 | 7 | 1 | 4 | 7 | 8 | 8 | 6 | 4 | 1 | 2 | 4 | 5 | 5 | 4 | 3 | 1 | 1 | 3 | 4 | 4 | 4 | 2 | 1 | 1 | 3 | 4 |
| 82 | 80 | 72 | 61 | 46 | 30 | 15 | 1 | 9 | 16 | 19 | 19 | 14 | 8 | 1 | 5 | 9 | 12 | 12 | 10 | 6 | 2 | 3 | 7 | 9 | 10 | 8 | 6 | 2 | 2 | 6 | 8 |
| 111 | 107 | 97 | 82 | 62 | 41 | 20 | 2 | 12 | 21 | 25 | 24 | 18 | 11 | 2 | 6 | 12 | 16 | 16 | 13 | 8 | 2 | 4 | 9 | 12 | 13 | 11 | 8 | 2 | 3 | 9 | 11 |
| 118 | 114 | 103 | 86 | 66 | 43 | 21 | 2 | 13 | 23 | 27 | 25 | 20 | 11 | 2 | 7 | 13 | 17 | 17 | 14 | 9 | 2 | 4 | 10 | 13 | 14 | 12 | 8 | 3 | 3 | 8 | 12 |
| 102 | 99 | 90 | 75 | 57 | 39 | 19 | 2 | 11 | 20 | 23 | 22 | 17 | 10 | 2 | 6 | 11 | 14 | 15 | 12 | 8 | 2 | 4 | 8 | 11 | 12 | 10 | 7 | 2 | 3 | 7 | 10 |
| 68 | 65 | 60 | 50 | 38 | 25 | 12 | 1 | 9 | 13 | 15 | 15 | 11 | 7 | 1 | 4 | 8 | 10 | 10 | 8 | 5 | 1 | 2 | 6 | 8 | 8 | 7 | 5 | 2 | 2 | 5 | 7 |
| 22 | 21 | 19 | 16 | 12 | 8 | 4 | 0 | 2 | 4 | 5 | 5 | 4 | 2 | 0 | 1 | 2 | 3 | 3 | 3 | 2 | 0 | 1 | 2 | 2 | 3 | 2 | 2 | 0 | 1 | 2 | 2 |
| 27 | 26 | 24 | 20 | 15 | 10 | 5 | 0 | 3 | 5 | 6 | 6 | 5 | 3 | 0 | 2 | 3 | 4 | 4 | 3 | 2 | 1 | 1 | 2 | 3 | 3 | 3 | 2 | 1 | 1 | 2 | 3 |
| 71 | 69 | 62 | 52 | 40 | 26 | 13 | 1 | 9 | 14 | 16 | 15 | 12 | 7 | 1 | 4 | [illegible] | 10 | 10 | 8 | 5 | 1 | 3 | 6 | 8 | 8 | 7 | 5 | 2 | 2 | 5 | 7 |
| 100 | 97 | 83 | 74 | 56 | 37 | 18 | 2 | 11 | 19 | 22 | 21 | 17 | 10 | 2 | 6 | 11 | 14 | 14 | 12 | 7 | 2 | 4 | 8 | 11 | 12 | 10 | 7 | 2 | 3 | 7 | 10 |
| 111 | 107 | 97 | 82 | 62 | 41 | 20 | 2 | 12 | 21 | 25 | 24 | 18 | 11 | 2 | 6 | 12 | 16 | 16 | 13 | 8 | 2 | 4 | 9 | 12 | 13 | 11 | 8 | 2 | 3 | 8 | 11 |
| 100 | 97 | 86 | 74 | 56 | 37 | 18 | 2 | 11 | 19 | 23 | 21 | 17 | 10 | 2 | 6 | 11 | 14 | 14 | 12 | 7 | 2 | 4 | 8 | 11 | 12 | 10 | 7 | 2 | 3 | 7 | 10 |
| 71 | 69 | 62 | 52 | 40 | 26 | 13 | 1 | 9 | 14 | 15 | 15 | 12 | 7 | 1 | 4 | 8 | 10 | 10 | 8 | 5 | 1 | 3 | 6 | 8 | 8 | 7 | 5 | 2 | 2 | 5 | 7 |
| 27 | 26 | 24 | 20 | 15 | 10 | 5 | 0 | 3 | 5 | 6 | 6 | 5 | 3 | 0 | 2 | 3 | 4 | 4 | 3 | 2 | 1 | 1 | 2 | 3 | 3 | 3 | 2 | 1 | 1 | 2 | 3 |
| 22 | 21 | 19 | 16 | 12 | 8 | 4 | 0 | 2 | 4 | 5 | 5 | 4 | 2 | 0 | 1 | 2 | 3 | 3 | 3 | 2 | 0 | 1 | 2 | 2 | 3 | 2 | 2 | 0 | 1 | 2 | 2 |
| 68 | 66 | 60 | 50 | 38 | 25 | 12 | 1 | 8 | 13 | 15 | 15 | 11 | 7 | 1 | 4 | 8 | 10 | 10 | 8 | 5 | 1 | 2 | 6 | 8 | 8 | 7 | 5 | 2 | 2 | 5 | 7 |
| 102 | 99 | 90 | 75 | 57 | 38 | 18 | 2 | 11 | 20 | 22 | 22 | 17 | 10 | 2 | 6 | 11 | 14 | 15 | 12 | 8 | 2 | 4 | 8 | 11 | 12 | 10 | 7 | 2 | 3 | 7 | 10 |
| 118 | 114 | 103 | 86 | 65 | 43 | 21 | 2 | 13 | 23 | 27 | 25 | 20 | 11 | 2 | 7 | 13 | 17 | 17 | 14 | 9 | 2 | 4 | 10 | 13 | 14 | 12 | 8 | 3 | 3 | 8 | 12 |
| 111 | 107 | 97 | 82 | 62 | 41 | 20 | 2 | 12 | 21 | 25 | 24 | 18 | 11 | 2 | 6 | 12 | 16 | 16 | 13 | 8 | 2 | 4 | 9 | 12 | 13 | 11 | 8 | 2 | 3 | 8 | 11 |
| 82 | 80 | 72 | 61 | 46 | 30 | 15 | 1 | 9 | 16 | 19 | 19 | 14 | 8 | 1 | 5 | 9 | 12 | 12 | 10 | 6 | 2 | 3 | 7 | 9 | 10 | 8 | 6 | 2 | 2 | 6 | 8 |
| 37 | 35 | 32 | 27 | 20 | 13 | 7 | 1 | 4 | 7 | 8 | 8 | 6 | 4 | 1 | 2 | 4 | 5 | 5 | 4 | 3 | 1 | 1 | 3 | 4 | 4 | 4 | 2 | 1 | 1 | 3 | 4 |
| 19 | 18 | 17 | 14 | 11 | 7 | 3 | 0 | 2 | 4 | 4 | 4 | 3 | 2 | 0 | 1 | 2 | 3 | 3 | 2 | 1 | 0 | 1 | 2 | 2 | 2 | 2 | 1 | 0 | 1 | 1 | 2 |
| 74 | 72 | 65 | 54 | 41 | 27 | 13 | 1 | 9 | 14 | 17 | 16 | 12 | 7 | 1 | 4 | 8 | 10 | 11 | 9 | 6 | 1 | 3 | 6 | 8 | 9 | 8 | 5 | 2 | 2 | 5 | 7 |
| 119 | 115 | 104 | 87 | 66 | 44 | 21 | 2 | 13 | 23 | 27 | 25 | 20 | 11 | 2 | 7 | 13 | 17 | 17 | 14 | 9 | 2 | 4 | 10 | 13 | 14 | 12 | 8 | 3 | 3 | 8 | 12 |
| 143 | 138 | 125 | 105 | 80 | 52 | 26 | 2 | 16 | 29 | 32 | 31 | 24 | 14 | 2 | 9 | 16 | 20 | 20 | 17 | 11 | 3 | 5 | 12 | 16 | 17 | 15 | 10 | 3 | 4 | 10 | 14 |
| 141 | 137 | 124 | 104 | 79 | 52 | 25 | 2 | 16 | 27 | 32 | 30 | 23 | 14 | 2 | 8 | 16 | 20 | 20 | 17 | 10 | 3 | 5 | 12 | 16 | 17 | 14 | 10 | 3 | 4 | 10 | 14 |
| 111 | 107 | 97 | 82 | 62 | 41 | 20 | 2 | 12 | 21 | 25 | 24 | 18 | 11 | 2 | 6 | 12 | 16 | 16 | 13 | 9 | 2 | 4 | 9 | 12 | 13 | 11 | 8 | 2 | 3 | 8 | 11 |
| 56 | 54 | 49 | 41 | 31 | 20 | 10 | 1 | 6 | 11 | 13 | 12 | 9 | 5 | 1 | 3 | 6 | 8 | 8 | 7 | 4 | 1 | 2 | 5 | 6 | 7 | 6 | 4 | 1 | 2 | 4 | 6 |
| 17 | 17 | 15 | 13 | 10 | 6 | 3 | 0 | 2 | 3 | 4 | 4 | 3 | 2 | 0 | 1 | 2 | 2 | 2 | 2 | 1 | 0 | 1 | 1 | 2 | 2 | 2 | 1 | 0 | 0 | 1 | 2 |
| 96 | 93 | 84 | 70 | 54 | 35 | 17 | 2 | 11 | 17 | 22 | 20 | 16 | 9 | 2 | 5 | 11 | 14 | 14 | 11 | 7 | 2 | 4 | 8 | 11 | 11 | 10 | 7 | 2 | 3 | 7 | 10 |
| 166 | 161 | 146 | 122 | 92 | 61 | 30 | 3 | 19 | 32 | 37 | 36 | 29 | 16 | 3 | 9 | 18 | 23 | 24 | 20 | 12 | 3 | 6 | 14 | 18 | 20 | 17 | 11 | 4 | 5 | 12 | 17 |
| 214 | 207 | 187 | 157 | 119 | 78 | 39 | 3 | 24 | 41 | 48 | 46 | 36 | 20 | 4 | 12 | 24 | 30 | 31 | 25 | 16 | 4 | 8 | 18 | 24 | 25 | 22 | 15 | 5 | 6 | 15 | 21 |
| 225 | 218 | 197 | 166 | 126 | 83 | 41 | 4 | 25 | 43 | 51 | 49 | 37 | 22 | 4 | 13 | 25 | 32 | 32 | 27 | 17 | 4 | 8 | 19 | 25 | 27 | 23 | 15 | 5 | 6 | 16 | 23 |

FT.LOG;1 20-JAN-1984 10:10 Page 7
**** TRANSFORMED DATA ****

**2. Modulus plane for transform of 8 by 8 rectangle**

FT.LOG;1 20-JAN-1984 10:10 Page 8
**** TRANSFORMED DATA ****
INPUT FILES REAL : SWFLI010 IMAGINARY : NULL0000
ANGLE (DEGREES) LEFT HALF SCALE FACTOR = 510.2751

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 73 | 146 | 219 | 293 | 6 | 79 | 152 | 45 | 118 | 191 | 264 | 338 | 51 | 124 | 17 | 90 | 163 | 236 | 309 | 22 | 96 | 349 | 62 | 135 | 208 | 281 | 354 | 67 | 321 | 34 | 107 |
| 73 | 146 | 219 | 293 | 6 | 79 | 152 | 225 | 118 | 191 | 264 | 338 | 51 | 124 | 197 | 90 | 163 | 236 | 309 | 23 | 96 | 169 | 62 | 135 | 208 | 281 | 354 | 67 | 141 | 34 | 107 | 180 |
| 146 | 219 | 293 | 6 | 79 | 152 | 225 | 298 | 191 | 264 | 338 | 51 | 124 | 197 | 270 | 163 | 236 | 309 | 23 | 96 | 169 | 242 | 135 | 208 | 281 | 354 | 67 | 141 | 214 | 107 | 180 | 253 |
| 219 | 293 | 6 | 79 | 152 | 225 | 298 | 11 | 264 | 338 | 51 | 124 | 197 | 270 | 343 | 236 | 309 | 23 | 96 | 169 | 242 | 315 | 208 | 281 | 354 | 68 | 141 | 214 | 287 | 180 | 253 | 326 |
| 293 | 6 | 79 | 152 | 225 | 298 | 11 | 84 | 338 | 51 | 124 | 197 | 270 | 343 | 56 | 309 | 23 | 96 | 169 | 242 | 315 | 28 | 281 | 354 | 68 | 141 | 214 | 287 | 0 | 253 | 325 | 39 |
| 6 | 79 | 152 | 225 | 298 | 11 | 84 | 158 | 51 | 124 | 197 | 270 | 343 | 56 | 129 | 22 | 96 | 169 | 242 | 315 | 28 | 101 | 354 | 68 | 141 | 214 | 287 | 0 | 73 | 326 | 39 | 113 |
| 79 | 152 | 225 | 298 | 11 | 84 | 158 | 231 | 124 | 197 | 270 | 343 | 56 | 129 | 203 | 36 | 169 | 242 | 315 | 28 | 101 | 174 | 68 | 141 | 214 | 287 | 0 | 73 | 146 | 39 | 113 | 186 |
| 152 | 225 | 298 | 11 | 84 | 158 | 231 | 304 | 197 | 270 | 343 | 56 | 129 | 203 | 276 | 169 | 242 | 315 | 28 | 101 | 174 | 248 | 141 | 214 | 287 | 0 | 73 | 146 | 219 | 113 | 186 | 259 |
| 45 | 118 | 191 | 264 | 338 | 51 | 124 | 197 | 90 | 163 | 236 | 309 | 23 | 96 | 169 | 62 | 135 | 208 | 281 | 354 | 67 | 141 | 34 | 107 | 180 | 253 | 326 | 39 | 112 | 6 | 79 | 152 |
| 118 | 191 | 264 | 338 | 51 | 124 | 197 | 270 | 163 | 236 | 309 | 23 | 96 | 169 | 242 | 135 | 208 | 281 | 354 | 68 | 141 | 214 | 107 | 180 | 253 | 326 | 39 | 112 | 186 | 79 | 152 | 225 |
| 191 | 264 | 338 | 51 | 124 | 197 | 270 | 343 | 236 | 309 | 23 | 96 | 169 | 242 | 315 | 208 | 281 | 354 | 68 | 141 | 214 | 287 | 180 | 253 | 326 | 39 | 112 | 186 | 259 | 152 | 225 | 298 |
| 264 | 338 | 51 | 124 | 197 | 270 | 343 | 56 | 309 | 23 | 96 | 169 | 242 | 315 | 28 | 281 | 354 | 68 | 141 | 214 | 287 | 0 | 253 | 326 | 39 | 113 | 186 | 259 | 332 | 225 | 298 | 11 |
| 338 | 51 | 124 | 197 | 270 | 343 | 56 | 129 | 22 | 96 | 169 | 242 | 315 | 28 | 101 | 354 | 68 | 141 | 214 | 287 | 0 | 73 | 326 | 39 | 113 | 186 | 259 | 332 | 45 | 298 | 11 | 84 |
| 51 | 124 | 197 | 270 | 343 | 56 | 129 | 203 | 96 | 169 | 242 | 315 | 28 | 101 | 174 | 67 | 141 | 214 | 287 | 0 | 73 | 146 | 39 | 113 | 186 | 259 | 332 | 45 | 118 | 11 | 84 | 158 |
| 124 | 197 | 270 | 343 | 56 | 129 | 203 | 276 | 169 | 242 | 315 | 28 | 101 | 174 | 249 | 141 | 214 | 287 | 0 | 73 | 146 | 219 | 113 | 186 | 259 | 332 | 45 | 118 | 191 | 84 | 158 | 231 |
| 17 | 90 | 163 | 236 | 309 | 22 | 96 | 169 | 62 | 135 | 208 | 281 | 354 | 67 | 141 | 34 | 107 | 180 | 253 | 326 | 39 | 112 | 6 | 79 | 152 | 225 | 298 | 11 | 84 | 337 | 51 | 124 |
| 90 | 163 | 236 | 309 | 23 | 96 | 169 | 242 | 135 | 208 | 281 | 354 | 68 | 141 | 214 | 107 | 180 | 253 | 326 | 39 | 112 | 186 | 79 | 152 | 225 | 298 | 11 | 84 | 157 | 51 | 124 | 197 |
| 163 | 236 | 309 | 23 | 96 | 169 | 242 | 315 | 208 | 281 | 354 | 68 | 141 | 214 | 287 | 180 | 253 | 326 | 39 | 113 | 186 | 259 | 152 | 225 | 298 | 11 | 84 | 157 | 231 | 124 | 197 | 270 |
| 236 | 309 | 23 | 96 | 169 | 242 | 315 | 28 | 281 | 354 | 68 | 141 | 214 | 287 | 0 | 253 | 326 | 39 | 112 | 186 | 259 | 332 | 225 | 298 | 11 | 84 | 157 | 231 | 304 | 197 | 270 | 343 |
| 309 | 22 | 96 | 169 | 242 | 315 | 28 | 101 | 354 | 68 | 141 | 214 | 297 | 0 | 73 | 326 | 39 | 112 | 186 | 259 | 332 | 45 | 298 | 11 | 84 | 157 | 231 | 304 | 17 | 270 | 343 | 56 |
| 22 | 96 | 169 | 242 | 315 | 28 | 101 | 174 | 67 | 141 | 214 | 287 | 0 | 73 | 146 | 39 | 112 | 186 | 259 | 332 | 45 | 118 | 11 | 84 | 157 | 231 | 304 | 17 | 90 | 343 | 56 | 129 |
| 96 | 169 | 242 | 315 | 28 | 101 | 174 | 248 | 141 | 214 | 287 | 0 | 73 | 146 | 219 | 112 | 186 | 259 | 332 | 45 | 119 | 191 | 84 | 157 | 231 | 304 | 17 | 90 | 163 | 56 | 129 | 202 |
| 349 | 62 | 135 | 208 | 231 | 354 | 68 | 141 | 34 | 107 | 180 | 253 | 326 | 39 | 113 | 6 | 79 | 152 | 225 | 298 | 11 | 84 | 338 | 51 | 124 | 197 | 270 | 343 | 56 | 309 | 23 | 96 |
| 62 | 135 | 208 | 281 | 354 | 68 | 141 | 214 | 107 | 180 | 253 | 326 | 39 | 113 | 186 | 79 | 152 | 225 | 298 | 11 | 84 | 157 | 51 | 124 | 197 | 270 | 343 | 56 | 129 | 23 | 96 | 169 |
| 135 | 208 | 281 | 354 | 68 | 141 | 214 | 287 | 180 | 253 | 326 | 39 | 113 | 186 | 259 | 152 | 225 | 298 | 11 | 84 | 157 | 231 | 124 | 197 | 270 | 343 | 56 | 129 | 202 | 96 | 169 | 242 |
| 208 | 281 | 354 | 68 | 141 | 214 | 287 | 0 | 253 | 326 | 39 | 113 | 186 | 259 | 332 | 225 | 298 | 11 | 84 | 158 | 231 | 304 | 197 | 270 | 343 | 56 | 129 | 202 | 276 | 169 | 242 | 315 |
| 281 | 354 | 67 | 141 | 214 | 287 | 0 | 73 | 326 | 39 | 112 | 186 | 259 | 332 | 45 | 298 | 11 | 84 | 157 | 231 | 304 | 17 | 270 | 343 | 56 | 129 | 202 | 276 | 349 | 242 | 315 | 28 |
| 354 | 67 | 141 | 214 | 287 | 0 | 73 | 146 | 39 | 112 | 186 | 259 | 332 | 45 | 118 | 11 | 84 | 157 | 231 | 304 | 17 | 90 | 343 | 56 | 129 | 202 | 276 | 349 | 62 | 315 | 28 | 101 |
| 67 | 141 | 214 | 287 | 0 | 73 | 146 | 219 | 112 | 186 | 259 | 332 | 45 | 118 | 191 | 84 | 157 | 231 | 304 | 17 | 90 | 163 | 56 | 129 | 202 | 276 | 349 | 62 | 135 | 28 | 101 | 174 |
| 321 | 34 | 107 | 180 | 253 | 326 | 39 | 113 | 6 | 79 | 152 | 225 | 298 | 11 | 84 | 338 | 51 | 124 | 197 | 270 | 343 | 56 | 309 | 23 | 95 | 169 | 242 | 315 | 28 | 231 | 354 | 68 |
| 34 | 107 | 180 | 253 | 326 | 39 | 113 | 186 | 79 | 152 | 225 | 298 | 11 | 84 | 158 | 51 | 124 | 197 | 270 | 343 | 56 | 129 | 22 | 96 | 169 | 242 | 315 | 28 | 101 | 354 | 68 | 141 |
| 107 | 180 | 253 | 326 | 39 | 113 | 186 | 259 | 152 | 225 | 298 | 11 | 84 | 158 | 231 | 124 | 197 | 270 | 343 | 56 | 129 | 202 | 96 | 169 | 242 | 315 | 28 | 101 | 174 | 68 | 141 | 214 |
| 180 | 253 | 326 | 39 | 113 | 186 | 259 | 332 | 225 | 298 | 11 | 84 | 158 | 231 | 304 | 197 | 270 | 343 | 56 | 129 | 202 | 276 | 169 | 242 | 315 | 28 | 101 | 174 | 247 | 141 | 214 | 287 |
| 253 | 326 | 39 | 113 | 186 | 259 | 332 | 45 | 298 | 11 | 84 | 158 | 231 | 304 | 17 | 270 | 343 | 56 | 129 | 203 | 276 | 349 | 242 | 315 | 28 | 101 | 174 | 247 | 321 | 214 | 287 | 0 |
| 326 | 39 | 113 | 186 | 259 | 332 | 45 | 118 | 11 | 84 | 158 | 231 | 304 | 17 | 90 | 343 | 56 | 129 | 203 | 276 | 349 | 62 | 315 | 28 | 101 | 174 | 247 | 321 | 34 | 287 | 0 | 73 |
| 39 | 113 | 186 | 259 | 332 | 45 | 118 | 191 | 84 | 158 | 231 | 304 | 17 | 90 | 163 | 56 | 129 | 202 | 276 | 349 | 62 | 135 | 28 | 101 | 174 | 248 | 321 | 34 | 107 | 0 | 73 | 146 |
| 292 | 6 | 79 | 152 | 225 | 298 | 11 | 84 | 338 | 51 | 124 | 197 | 270 | 343 | 56 | 309 | 22 | 96 | 169 | 242 | 315 | 28 | 281 | 354 | 68 | 141 | 214 | 287 | 0 | 253 | 326 | 39 |
| 6 | 79 | 152 | 225 | 298 | 11 | 84 | 158 | 51 | 124 | 197 | 270 | 343 | 56 | 129 | 22 | 96 | 169 | 242 | 315 | 28 | 101 | 354 | 68 | 141 | 214 | 287 | 0 | 73 | 326 | 39 | 113 |
| 79 | 152 | 225 | 298 | 11 | 84 | 158 | 231 | 124 | 197 | 270 | 343 | 56 | 129 | 203 | 96 | 169 | 242 | 315 | 28 | 101 | 174 | 68 | 141 | 214 | 287 | 0 | 73 | 146 | 39 | 113 | 186 |
| 152 | 225 | 298 | 11 | 84 | 158 | 231 | 304 | 197 | 270 | 343 | 56 | 129 | 203 | 276 | 169 | 242 | 315 | 28 | 101 | 174 | 247 | 141 | 214 | 287 | 0 | 73 | 146 | 219 | 113 | 186 | 259 |
| 225 | 298 | 11 | 84 | 158 | 231 | 304 | 17 | 270 | 343 | 56 | 129 | 203 | 276 | 349 | 242 | 315 | 28 | 101 | 174 | 247 | 321 | 214 | 287 | 0 | 73 | 146 | 219 | 292 | 186 | 259 | 332 |
| 298 | 11 | 84 | 158 | 231 | 304 | 17 | 90 | 343 | 56 | 129 | 203 | 276 | 349 | 62 | 315 | 28 | 101 | 174 | 248 | 321 | 34 | 287 | 0 | 73 | 146 | 219 | 292 | 6 | 259 | 332 | 45 |
| 11 | 84 | 158 | 231 | 304 | 17 | 90 | 163 | 56 | 129 | 203 | 276 | 349 | 62 | 135 | 28 | 101 | 174 | 248 | 321 | 34 | 107 | 0 | 73 | 146 | 219 | 293 | 6 | 79 | 332 | 45 | 118 |
| 264 | 338 | 51 | 124 | 197 | 270 | 343 | 56 | 309 | 23 | 96 | 169 | 242 | 315 | 23 | 281 | 354 | 68 | 141 | 214 | 287 | 0 | 253 | 326 | 39 | 113 | 186 | 259 | 332 | 225 | 298 | 11 |
| 338 | 51 | 124 | 197 | 270 | 343 | 56 | 129 | 23 | 96 | 169 | 242 | 315 | 28 | 101 | 354 | 68 | 141 | 214 | 287 | 0 | 73 | 326 | 39 | 113 | 186 | 259 | 332 | 45 | 298 | 11 | 84 |
| 51 | 124 | 197 | 270 | 343 | 56 | 129 | 203 | 96 | 169 | 242 | 315 | 28 | 101 | 174 | 67 | 141 | 214 | 287 | 0 | 73 | 146 | 39 | 113 | 186 | 259 | 332 | 45 | 118 | 11 | 84 | 158 |
| 124 | 197 | 270 | 343 | 56 | 129 | 203 | 276 | 169 | 242 | 315 | 28 | 101 | 174 | 249 | 141 | 214 | 287 | 0 | 73 | 146 | 219 | 113 | 186 | 259 | 332 | 45 | 118 | 191 | 84 | 158 | 231 |
| 197 | 270 | 343 | 56 | 129 | 203 | 276 | 349 | 242 | 315 | 28 | 101 | 174 | 248 | 321 | 214 | 287 | 0 | 73 | 146 | 219 | 292 | 186 | 259 | 332 | 45 | 118 | 191 | 264 | 158 | 231 | 304 |
| 270 | 343 | 56 | 129 | 203 | 276 | 349 | 62 | 315 | 28 | 101 | 174 | 248 | 321 | 34 | 287 | 0 | 73 | 146 | 219 | 292 | 6 | 259 | 332 | 45 | 118 | 191 | 264 | 337 | 231 | 304 | 17 |
| 343 | 56 | 129 | 203 | 276 | 349 | 62 | 135 | 28 | 101 | 174 | 248 | 321 | 34 | 107 | 0 | 73 | 146 | 219 | 293 | 6 | 79 | 332 | 45 | 118 | 191 | 264 | 337 | 51 | 304 | 17 | 90 |
| 236 | 309 | 22 | 96 | 169 | 242 | 315 | 23 | 281 | 354 | 67 | 141 | 214 | 287 | 0 | 253 | 326 | 39 | 112 | 186 | 259 | 332 | 225 | 298 | 11 | 84 | 157 | 231 | 304 | 197 | 270 | 343 |
| 309 | 22 | 96 | 169 | 242 | 315 | 28 | 101 | 354 | 67 | 141 | 214 | 287 | 0 | 73 | 326 | 39 | 112 | 186 | 259 | 332 | 45 | 298 | 11 | 84 | 157 | 231 | 304 | 17 | 270 | 343 | 56 |
| 23 | 96 | 169 | 242 | 315 | 28 | 101 | 174 | 68 | 141 | 214 | 287 | 0 | 73 | 146 | 39 | 113 | 186 | 259 | 332 | 45 | 118 | 11 | 84 | 158 | 231 | 304 | 17 | 90 | 343 | 56 | 129 |
| 96 | 169 | 242 | 315 | 28 | 101 | 174 | 248 | 141 | 214 | 287 | 0 | 73 | 146 | 219 | 112 | 186 | 259 | 332 | 45 | 118 | 191 | 84 | 158 | 231 | 304 | 17 | 90 | 163 | 56 | 129 | 203 |
| 169 | 242 | 315 | 28 | 101 | 174 | 248 | 321 | 214 | 287 | 0 | 73 | 146 | 219 | 293 | 186 | 259 | 332 | 45 | 118 | 191 | 264 | 158 | 231 | 304 | 17 | 90 | 163 | 236 | 129 | 203 | 276 |

FT.LOG;1 20-JAN-1984 10:10 Page 9
**** TRANSFORMED DATA ****

3. Phase plane for transform of 8 by 8 test object at position 10, 10

,QFT02.LOG;1 14-NOV-1983 14:52 Page 40

**** TRANSFORMED DATA ****

INPUT FILES REAL : SQ202 IMAGINARY : NULL

ANGLE (DEGREES) LEFT HALF SCALE FACTOR = 510.2761

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 129 | 259 | 28 | 158 | 287 | 56 | 186 | 135 | 264 | 34 | 163 | 293 | 62 | 191 | 141 | 270 | 39 | 169 | 298 | 68 | 197 | 146 | 276 | 45 | 174 | 304 | 73 | 202 | 152 | 281 | 51 |
| 129 | 259 | 28 | 158 | 287 | 56 | 186 | 315 | 264 | 34 | 163 | 293 | 62 | 191 | 321 | 270 | 39 | 169 | 298 | 68 | 197 | 326 | 276 | 45 | 174 | 304 | 73 | 203 | 332 | 281 | 51 | 180 |
| 259 | 28 | 158 | 287 | 56 | 186 | 315 | 84 | 34 | 163 | 293 | 62 | 191 | 321 | 90 | 39 | 169 | 298 | 68 | 197 | 326 | 96 | 45 | 174 | 304 | 73 | 203 | 332 | 101 | 51 | 180 | 309 |
| 28 | 158 | 287 | 56 | 186 | 315 | 84 | 214 | 163 | 293 | 62 | 191 | 321 | 90 | 219 | 169 | 298 | 68 | 197 | 326 | 96 | 225 | 174 | 304 | 73 | 203 | 332 | 101 | 231 | 180 | 309 | 79 |
| 158 | 287 | 56 | 186 | 315 | 84 | 214 | 343 | 293 | 62 | 191 | 321 | 90 | 219 | 349 | 298 | 68 | 197 | 326 | 96 | 225 | 354 | 304 | 73 | 203 | 332 | 101 | 231 | 360 | 309 | 79 | 208 |
| 287 | 56 | 186 | 315 | 84 | 214 | 343 | 113 | 62 | 191 | 321 | 90 | 219 | 349 | 118 | 68 | 197 | 326 | 96 | 225 | 354 | 124 | 73 | 203 | 332 | 101 | 231 | 0 | 129 | 79 | 208 | 338 |
| 56 | 186 | 315 | 84 | 214 | 343 | 113 | 242 | 191 | 321 | 90 | 219 | 349 | 118 | 247 | 197 | 326 | 96 | 225 | 354 | 124 | 253 | 202 | 332 | 101 | 231 | 0 | 129 | 259 | 208 | 338 | 107 |
| 186 | 315 | 84 | 214 | 343 | 113 | 242 | 11 | 321 | 90 | 219 | 349 | 118 | 247 | 17 | 326 | 96 | 225 | 354 | 124 | 253 | 23 | 332 | 101 | 231 | 0 | 129 | 259 | 28 | 338 | 107 | 236 |
| 135 | 264 | 34 | 163 | 293 | 62 | 191 | 321 | 270 | 39 | 169 | 298 | 68 | 197 | 326 | 276 | 45 | 174 | 304 | 73 | 203 | 332 | 281 | 51 | 180 | 309 | 79 | 208 | 338 | 287 | 56 | 186 |
| 264 | 34 | 163 | 293 | 62 | 191 | 321 | 90 | 39 | 169 | 298 | 68 | 197 | 326 | 96 | 45 | 174 | 304 | 73 | 203 | 332 | 101 | 51 | 180 | 309 | 79 | 208 | 338 | 107 | 56 | 186 | 315 |
| 34 | 163 | 293 | 62 | 191 | 321 | 90 | 219 | 169 | 298 | 68 | 197 | 326 | 96 | 225 | 174 | 304 | 73 | 203 | 332 | 101 | 231 | 180 | 309 | 79 | 208 | 338 | 107 | 236 | 186 | 315 | 84 |
| 163 | 293 | 62 | 191 | 321 | 90 | 219 | 349 | 298 | 68 | 197 | 326 | 96 | 225 | 354 | 304 | 73 | 203 | 332 | 101 | 231 | 0 | 309 | 79 | 208 | 338 | 107 | 236 | 6 | 315 | 84 | 214 |
| 292 | 62 | 191 | 321 | 90 | 219 | 349 | 118 | 67 | 197 | 326 | 96 | 225 | 354 | 124 | 73 | 202 | 332 | 101 | 231 | 0 | 129 | 79 | 208 | 338 | 107 | 236 | 6 | 135 | 84 | 214 | 343 |
| 62 | 191 | 321 | 90 | 219 | 349 | 118 | 247 | 197 | 326 | 96 | 225 | 354 | 124 | 253 | 203 | 332 | 101 | 231 | 360 | 129 | 259 | 208 | 337 | 107 | 236 | 6 | 135 | 264 | 214 | 343 | 112 |
| 191 | 321 | 90 | 219 | 349 | 118 | 247 | 17 | 326 | 96 | 225 | 354 | 124 | 253 | 22 | 332 | 101 | 231 | 360 | 129 | 259 | 28 | 337 | 107 | 236 | 6 | 135 | 264 | 34 | 343 | 112 | 242 |
| 141 | 270 | 39 | 169 | 298 | 68 | 197 | 326 | 276 | 45 | 174 | 304 | 73 | 203 | 332 | 281 | 51 | 180 | 309 | 79 | 208 | 338 | 287 | 56 | 186 | 315 | 84 | 214 | 343 | 293 | 62 | 191 |
| 270 | 39 | 169 | 298 | 68 | 197 | 326 | 96 | 45 | 174 | 304 | 73 | 202 | 332 | 101 | 51 | 180 | 309 | 79 | 208 | 338 | 107 | 56 | 186 | 315 | 84 | 214 | 343 | 112 | 62 | 191 | 321 |
| 39 | 169 | 298 | 68 | 197 | 326 | 96 | 225 | 174 | 304 | 73 | 203 | 332 | 101 | 231 | 180 | 309 | 79 | 208 | 338 | 107 | 236 | 186 | 315 | 84 | 214 | 343 | 113 | 242 | 191 | 321 | 90 |
| 169 | 298 | 68 | 197 | 326 | 96 | 225 | 354 | 304 | 73 | 203 | 332 | 101 | 231 | 360 | 309 | 79 | 208 | 338 | 107 | 236 | 6 | 315 | 84 | 214 | 343 | 113 | 242 | 11 | 321 | 90 | 219 |
| 298 | 68 | 197 | 326 | 96 | 225 | 354 | 124 | 73 | 203 | 332 | 101 | 231 | 360 | 129 | 79 | 208 | 338 | 107 | 236 | 6 | 135 | 84 | 214 | 343 | 113 | 242 | 11 | 141 | 90 | 219 | 349 |
| 68 | 197 | 326 | 96 | 225 | 354 | 124 | 253 | 203 | 332 | 101 | 231 | 0 | 129 | 259 | 208 | 338 | 107 | 236 | 6 | 135 | 264 | 214 | 343 | 113 | 242 | 11 | 141 | 270 | 219 | 349 | 118 |
| 197 | 326 | 96 | 225 | 354 | 124 | 253 | 23 | 332 | 101 | 231 | 0 | 129 | 259 | 28 | 338 | 107 | 236 | 6 | 135 | 264 | 34 | 343 | 113 | 242 | 11 | 141 | 270 | 39 | 349 | 118 | 248 |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 51 | 180 | 309 | 79 | 208 | 337 | 237 | 56 | 186 | 315 | 84 | 214 | 343 | 292 | 62 | 191 | 321 | 90 | 219 | 349 | 298 | 68 | 197 |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| 174 | 304 | 73 | 203 | 332 | 101 | 231 | 0 | 309 | 79 | 208 | 338 | 107 | 236 | 6 | 315 | 94 | 214 | 343 | 113 | 242 | 11 | 321 | 90 | 219 | 349 | [illegible] | 243 | 17 | 326 | 96 | 225 |
| 304 | 73 | 203 | 332 | 101 | 231 | 0 | 129 | 79 | 208 | 336 | 107 | 236 | 6 | 135 | 84 | 214 | 343 | 113 | 242 | 11 | 141 | 90 | 219 | 349 | 113 | 245 | 17 | 146 | 96 | 225 | 354 |
| 73 | 203 | 332 | 101 | 231 | 0 | 129 | 259 | 208 | 338 | 107 | 236 | 6 | 135 | 264 | 214 | 343 | 113 | 242 | 11 | 141 | 270 | 219 | 349 | 118 | 248 | 17 | 146 | 276 | 225 | 354 | 124 |
| 202 | 332 | 101 | 231 | 360 | 129 | 259 | 28 | 337 | 107 | 236 | 6 | 135 | 264 | 34 | 343 | 113 | 242 | 11 | 141 | 270 | 39 | 349 | 118 | 247 | 17 | 146 | 276 | 45 | 354 | 124 | 253 |
| 152 | 281 | 51 | 180 | 309 | 79 | 208 | 338 | 287 | 56 | 186 | 315 | 84 | 214 | 343 | 293 | 62 | 191 | 321 | 90 | 219 | 349 | 298 | 68 | 197 | 326 | 96 | 225 | 354 | 304 | 73 | 203 |
| 281 | 51 | 180 | 309 | 79 | 208 | 338 | 107 | 56 | 186 | 315 | 84 | 214 | 343 | 112 | 62 | 191 | 321 | 90 | 219 | 349 | 118 | 68 | 197 | 326 | 96 | 225 | 354 | 124 | 73 | 203 | 332 |
| 51 | 180 | 309 | 79 | 208 | 338 | 107 | 236 | 186 | 315 | 84 | 214 | 343 | 112 | 242 | 191 | 321 | 90 | 219 | 349 | 118 | 248 | 197 | 326 | 96 | 225 | 354 | 124 | 253 | 203 | 332 | 101 |
| 180 | 309 | 79 | 208 | 338 | 107 | 236 | 6 | 315 | 84 | 214 | 343 | 112 | 242 | 11 | 321 | 90 | 219 | 349 | 118 | 248 | 17 | 326 | 96 | 225 | 354 | 124 | 253 | 22 | 332 | 101 | 231 |
| 309 | 79 | 208 | 338 | 107 | 236 | 6 | 135 | 84 | 214 | 343 | 112 | 242 | 11 | 141 | 90 | 219 | 349 | 118 | 248 | 17 | 146 | 96 | 225 | 354 | 124 | 253 | 23 | 152 | 101 | 231 | 0 |
| 79 | 208 | 338 | 107 | 236 | 6 | 135 | 264 | 214 | 343 | 113 | 242 | 11 | 141 | 270 | 219 | 349 | 118 | 248 | 17 | 146 | 276 | 225 | 354 | 124 | 253 | 23 | 152 | 281 | 231 | 0 | 129 |
| 208 | 338 | 107 | 236 | 6 | 135 | 264 | 34 | 343 | 113 | 242 | 11 | 141 | 270 | 39 | 349 | 118 | 248 | 17 | 146 | 276 | 45 | 354 | 124 | 253 | 23 | 152 | 281 | 51 | 0 | 129 | 259 |
| 157 | 287 | 56 | 186 | 315 | 84 | 214 | 343 | 292 | 62 | 191 | 321 | 90 | 219 | 349 | 298 | 67 | 197 | 326 | 96 | 225 | 354 | 304 | 73 | 202 | 332 | 101 | 231 | 360 | 309 | 79 | 208 |
| 287 | 56 | 186 | 315 | 84 | 214 | 343 | 112 | 62 | 191 | 321 | 90 | 219 | 349 | 118 | 68 | 197 | 326 | 96 | 225 | 354 | 124 | 73 | 202 | 332 | 101 | 231 | 0 | 129 | 79 | 208 | 338 |
| 56 | 186 | 315 | 84 | 214 | 343 | 113 | 242 | 191 | 321 | 90 | 219 | 349 | 118 | 247 | 197 | 326 | 96 | 225 | 354 | 124 | 253 | 202 | 332 | 101 | 231 | 0 | 129 | 259 | 208 | 338 | 107 |
| 186 | 315 | 84 | 214 | 343 | 113 | 242 | 11 | 321 | 90 | 219 | 349 | 118 | 247 | 17 | 326 | 96 | 225 | 354 | 124 | 253 | 23 | 332 | 101 | 231 | 0 | 129 | 259 | 28 | 338 | 107 | 236 |
| 315 | 84 | 214 | 343 | 113 | 242 | 11 | 141 | 90 | 219 | 349 | 118 | 248 | 17 | 146 | 96 | 225 | 354 | 124 | 253 | 23 | 152 | 101 | 231 | 0 | 129 | 259 | 28 | 157 | 107 | 236 | 6 |
| 84 | 214 | 343 | 113 | 242 | 11 | 141 | 270 | 219 | 349 | 118 | 248 | 17 | 146 | 276 | 225 | 354 | 124 | 253 | 23 | 152 | 281 | 231 | 0 | 129 | 259 | 28 | 158 | 287 | 236 | 6 | 135 |
| 214 | 343 | 113 | 242 | 11 | 141 | 270 | 39 | 349 | 118 | 248 | 17 | 146 | 276 | 45 | 354 | 124 | 253 | 23 | 152 | 281 | 51 | 360 | 129 | 259 | 28 | 158 | 287 | 56 | 6 | 135 | 264 |
| 163 | 292 | 62 | 191 | 321 | 90 | 219 | 349 | 298 | 67 | 197 | 326 | 96 | 225 | 354 | 304 | 73 | 203 | 332 | 101 | 231 | 0 | 309 | 79 | 208 | 337 | 107 | 236 | 6 | 315 | 84 | 214 |
| 293 | 62 | 191 | 321 | 90 | 219 | 349 | 118 | 68 | 197 | 326 | 96 | 225 | 354 | 124 | 73 | 203 | 332 | 101 | 231 | 0 | 129 | 79 | 208 | 338 | 107 | 236 | 6 | 135 | 84 | 214 | 343 |
| 62 | 191 | 321 | 90 | 219 | 349 | 118 | 248 | 197 | 326 | 96 | 225 | 354 | 124 | 253 | 203 | 332 | 101 | 231 | 0 | 129 | 259 | 208 | 338 | 107 | 236 | 6 | 135 | 264 | 214 | 343 | 113 |
| 191 | 321 | 90 | 219 | 349 | 118 | 248 | 17 | 326 | 96 | 225 | 354 | 124 | 253 | 22 | 332 | 101 | 231 | 0 | 129 | 259 | 28 | 337 | 107 | 236 | 6 | 135 | 264 | 34 | 343 | 113 | 242 |
| 321 | 90 | 219 | 349 | 118 | 248 | 17 | 146 | 96 | 225 | 354 | 124 | 253 | 22 | 152 | 101 | 231 | 0 | 129 | 259 | 28 | 158 | 107 | 236 | 6 | 135 | 264 | 34 | 163 | 113 | 242 | 11 |
| 90 | 219 | 349 | 118 | 248 | 17 | 146 | 276 | 225 | 354 | 124 | 253 | 22 | 152 | 281 | 231 | 0 | 129 | 259 | 28 | 158 | 287 | 236 | 6 | 135 | 264 | 34 | 163 | 293 | 242 | 11 | 141 |
| 219 | 349 | 118 | 248 | 17 | 146 | 276 | 45 | 354 | 124 | 253 | 23 | 152 | 281 | 51 | 0 | 129 | 259 | 28 | 158 | 287 | 56 | 6 | 135 | 264 | 34 | 163 | 293 | 62 | 11 | 141 | 270 |
| 169 | 298 | 68 | 197 | 326 | 96 | 225 | 354 | 304 | 73 | 203 | 332 | 101 | 231 | 360 | 309 | 79 | 208 | 338 | 107 | 236 | 6 | 315 | 84 | 214 | 343 | 113 | 242 | 11 | 321 | 90 | 219 |
| 298 | 68 | 197 | 326 | 96 | 225 | 354 | 124 | 73 | 203 | 332 | 101 | 231 | 360 | 129 | 79 | 208 | 338 | 107 | 236 | 6 | 135 | 84 | 214 | 343 | 113 | 242 | 11 | 141 | 90 | 219 | 349 |
| 68 | 197 | 326 | 96 | 225 | 354 | 124 | 253 | 203 | 332 | 101 | 231 | 360 | 129 | 259 | 208 | 338 | 107 | 236 | 6 | 135 | 264 | 214 | 343 | 113 | 242 | 11 | 141 | 270 | 219 | 349 | 118 |
| 197 | 326 | 96 | 225 | 354 | 124 | 253 | 23 | 332 | 101 | 231 | 0 | 129 | 259 | 28 | 338 | 107 | 236 | 6 | 135 | 264 | 34 | 343 | 113 | 242 | 11 | 141 | 270 | 39 | 349 | 118 | 248 |
| 326 | 96 | 225 | 354 | 124 | 253 | 23 | 152 | 101 | 231 | 0 | 129 | 259 | 28 | 157 | 107 | 236 | 6 | 135 | 264 | 34 | 163 | 112 | 242 | 11 | 141 | 270 | 39 | 169 | 118 | 248 | 17 |

**4. Phase plane for transform of 8 by 8 test object at position 20, 20**

OG:7 25-JAN-1984 16:13 Page 2

**** INPUT DATA ****

INPUT FILES REAL : PI04THJP IMAGINARY : NULL0000

REAL PART LEFT HALF SCALE FACTOR = 6.2438

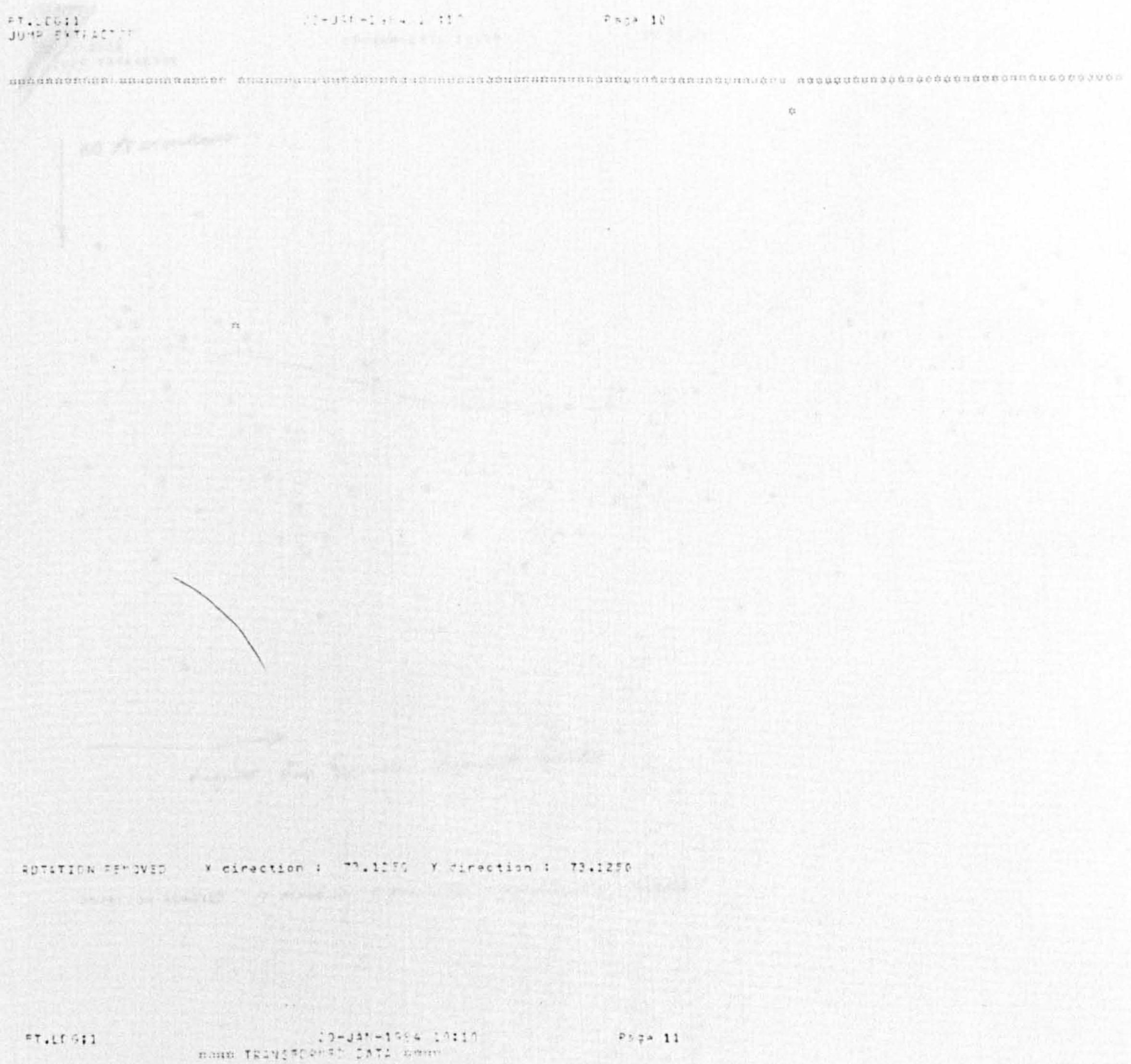
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 431 | 462 | 418 | 431 | 456 | 418 | 418 | 400 | 418 | 456 | 481 | 431 | 462 | 431 | 400 | 443 | 462 | 431 | 456 | 418 | 387 | 431 | 431 | 418 | 418 | 462 | 456 | 456 | 443 | 468 | 443 | 468 |
| 468 | 456 | 431 | 531 | 456 | 481 | 481 | 468 | 468 | 456 | 481 | 481 | 456 | 468 | 468 | 443 | 456 | 468 | 456 | 431 | 456 | 443 | 456 | 512 | 462 | 512 | 456 | 456 | 468 | 431 | 468 | 468 |
| 462 | 462 | 462 | 462 | 443 | 493 | 462 | 468 | 468 | 512 | 443 | 468 | 468 | 406 | 462 | 462 | 418 | 456 | 468 | 456 | 468 | 468 | 531 | 456 | 462 | 531 | 468 | 493 | 543 | 468 | 456 | 512 |
| 462 | 418 | 418 | 443 | 406 | 418 | 462 | 443 | 468 | 468 | 468 | 462 | 462 | 462 | 443 | 468 | 462 | 462 | 462 | 462 | 443 | 456 | 468 | 418 | 462 | 456 | 462 | 468 | 468 | 462 | 418 | 493 |
| 431 | 431 | 418 | 418 | 443 | 418 | 400 | 406 | 418 | 462 | 418 | 456 | 462 | 456 | 456 | 462 | 456 | 418 | 418 | 418 | 418 | 418 | 431 | 456 | 418 | 462 | 462 | 462 | 406 | 462 | 443 | 431 |
| 418 | 468 | 462 | 431 | 418 | 418 | 387 | 418 | 456 | 456 | 443 | 468 | 418 | 443 | 456 | 406 | 443 | 418 | 443 | 406 | 431 | 443 | 431 | 431 | 418 | 431 | 418 | 431 | 431 | 418 | 431 | 418 |
| 418 | 406 | 456 | 325 | 387 | 443 | 368 | 387 | 456 | 381 | 406 | 468 | 387 | 418 | 531 | 375 | 418 | 481 | 356 | 375 | 468 | 362 | 418 | 468 | 368 | 406 | 493 | 387 | 462 | 481 | 406 | 431 |
| 431 | 456 | 418 | 418 | 431 | 418 | 381 | 431 | 456 | 400 | 462 | 531 | 387 | 512 | 406 | 387 | 468 | 406 | 362 | 456 | 418 | 381 | 456 | 406 | 456 | 431 | 387 | 468 | 456 | 393 | 418 | 493 |
| 462 | 443 | 418 | 400 | 418 | 387 | 406 | 393 | 406 | 406 | 418 | 418 | 418 | 393 | 400 | 418 | 406 | 406 | 375 | 418 | 387 | 381 | 387 | 387 | 393 | 418 | 387 | 400 | 400 | 418 | 393 | 418 |
| 418 | 400 | 400 | 387 | 418 | 393 | 393 | 418 | 418 | 381 | 418 | 481 | 456 | 387 | 431 | 387 | 387 | 387 | 350 | 362 | 387 | 368 | 368 | 387 | 381 | 362 | 387 | 381 | 387 | 400 | 387 | 381 |
| 418 | 387 | 375 | 368 | 393 | 343 | 418 | 418 | 350 | 387 | 381 | 318 | 375 | 362 | 368 | 381 | 368 | 381 | 381 | 325 | 381 | 325 | 356 | 393 | 362 | 400 | 393 | 387 | 368 | 418 | 362 | 368 |
| 400 | 387 | 418 | 418 | 393 | 418 | 387 | 387 | 387 | 331 | 343 | 387 | 350 | 331 | 387 | 381 | 356 | 387 | 356 | 350 | 381 | 368 | 368 | 381 | 381 | 381 | 387 | 368 | 381 | 400 | 418 | 387 |
| 368 | 393 | 368 | 381 | 393 | 375 | 362 | 406 | 325 | 343 | 381 | 318 | 350 | 368 | 331 | 350 | 400 | 331 | 387 | 387 | 325 | 381 | 406 | 331 | 368 | 387 | 368 | 375 | 387 | 362 | 387 | 362 |
| 350 | 318 | 325 | 331 | 325 | 331 | 350 | 325 | 318 | 325 | 318 | 0 | 318 | 331 | 331 | 325 | 350 | 331 | 350 | 350 | 325 | 343 | 350 | 350 | 350 | 331 | 325 | 343 | 331 | 350 | 325 | 350 |
| 368 | 325 | 343 | 356 | 343 | 331 | 350 | 350 | 325 | 325 | 312 | 343 | 343 | 325 | 325 | 350 | 350 | 325 | 350 | 0 | 325 | 350 | 350 | 356 | 368 | 343 | 325 | 325 | 350 | 343 | 318 | 331 |
| 368 | 343 | 312 | 343 | 331 | 325 | 350 | 325 | 325 | 318 | 325 | 325 | 325 | 331 | 312 | 350 | 318 | 0 | 0 | 0 | 318 | 312 | 0 | 325 | 325 | 0 | 0 | 318 | 0 | 0 | 0 | 0 |
| 325 | 350 | 0 | 0 | 356 | 312 | 0 | 331 | 0 | 0 | 325 | 0 | 0 | 331 | 0 | 0 | 331 | 0 | 0 | 0 | 0 | 0 | 318 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 331 | 368 | 0 | 343 | 368 | 0 | 350 | 312 | 0 | 312 | 0 | 0 | 325 | 0 | 0 | 331 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 312 | 312 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 312 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 331 | 375 | 574 | 631 | 606 | 705 | 705 | 624 | 568 | 406 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 387 | 531 | 568 | 674 | 649 | 631 | 649 | 549 | 393 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 456 | 587 | 606 | 755 | 743 | 743 | 843 | 799 | 712 | 687 | 456 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 325 | 687 | 731 | 724 | 918 | 743 | 843 | 999 | 868 | 755 | 724 | 481 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 312 | 368 | 737 | 799 | 824 | 949 | 918 | 855 | 918 | 974 | 880 | 780 | 512 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 312 | 456 | 731 | 843 | 824 | 843 | 855 | 868 | 868 | 843 | 780 | 718 | 493 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 418 | 737 | 793 | 824 | 837 | 868 | 855 | 837 | 843 | 755 | 743 | 481 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 368 | 687 | 731 | 731 | 843 | 824 | 824 | 824 | 780 | 724 | 649 | 481 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 312 | 531 | 587 | 624 | 731 | 724 | 718 | 749 | 687 | 674 | 618 | 624 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 387 | 418 | 543 | 631 | 624 | 631 | 643 | 631 | 624 | 574 | 531 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 343 | 368 | 387 | 493 | 418 | 418 | 481 | 462 | 443 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 312 | 325 | 387 | 418 | 406 | 443 | 462 | 443 | 400 | 456 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 312 | 387 | 343 | 362 | 381 | 406 | 325 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 325 | 325 | 0 | 0 | 325 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 312 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**5 a). Thresholded Jeep image from Alabama database (left half)**

```
 ;7                          25-JAN-1984 16:13                 Page 3
                          **** INPUT DATA ****
 INPUT FILES      REAL : PI04THJP      IMAGINARY : NULL0000
 REAL PART             RIGHT HALF         SCALE FACTOR =      6.2438
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 456 | 443 | 468 | 481 | 443 | 468 | 481 | 443 | 456 | 443 | 431 | 512 | 406 | 400 | 393 | 431 | 456 | 431 | 431 | 406 | 387 | 418 | 393 | 393 | 456 | 418 | 387 | 456 | 418 | 381 | 531 | 393 |
| 462 | 462 | 456 | 481 | 468 | 431 | 481 | 493 | 456 | 468 | 468 | 418 | 456 | 443 | 393 | 462 | 456 | 431 | 468 | 468 | 387 | 418 | 431 | 400 | 418 | 418 | 387 | 406 | 456 | 387 | 381 | 400 |
| 443 | 400 | 531 | 456 | 418 | 531 | 456 | 418 | 549 | 418 | 418 | 568 | 418 | 418 | 418 | 443 | 418 | 462 | 456 | 400 | 406 | 400 | 393 | 418 | 406 | 418 | 456 | 393 | 387 | 393 | 387 | 343 |
| 431 | 456 | 418 | 443 | 406 | 418 | 418 | 406 | 443 | 418 | 443 | 443 | 443 | 418 | 443 | 406 | 393 | 406 | 418 | 406 | 406 | 406 | 393 | 406 | 406 | 406 | 393 | 393 | 387 | 406 | 375 | 375 |
| 456 | 418 | 418 | 431 | 400 | 393 | 406 | 393 | 393 | 381 | 418 | 418 | 387 | 387 | 406 | 387 | 375 | 381 | 393 | 362 | 431 | 387 | 387 | 393 | 375 | 418 | 443 | 400 | 387 | 381 | 331 | 362 |
| 418 | 431 | 406 | 387 | 406 | 387 | 387 | 387 | 393 | 400 | 400 | 418 | 406 | 387 | 387 | 393 | 387 | 393 | 418 | 418 | 418 | 418 | 393 | 406 | 387 | 418 | 431 | 387 | 400 | 387 | 368 | 362 |
| 462 | 393 | 418 | 418 | 375 | 418 | 381 | 356 | 418 | 387 | 406 | 418 | 456 | 387 | 362 | 418 | 375 | 331 | 481 | 406 | 350 | 481 | 400 | 350 | 400 | 400 | 418 | 406 | 393 | 418 | 387 | 381 |
| 418 | 387 | 456 | 387 | 362 | 325 | 0 | 406 | 381 | 387 | 356 | 443 | 387 | 387 | 468 | 356 | 368 | 431 | 393 | 400 | 468 | 387 | 387 | 531 | 356 | 387 | 531 | 418 | 387 | 468 | 387 | 368 |
| 393 | 406 | 387 | 362 | 325 | 0 | 0 | 0 | 325 | 318 | 368 | 387 | 387 | 418 | 387 | 393 | 418 | 381 | 400 | 406 | 400 | 443 | 418 | 418 | 393 | 406 | 400 | 381 | 418 | 387 | 393 | 387 |
| 393 | 387 | 381 | 375 | 0 | 0 | 0 | 0 | 0 | 356 | 387 | 400 | 400 | 393 | 362 | 418 | 400 | 393 | 406 | 418 | 418 | 418 | 443 | 393 | 400 | 400 | 387 | 400 | 400 | 381 | 362 | 375 |
| 387 | 387 | 418 | 350 | 0 | 0 | 0 | 0 | 0 | 350 | 350 | 387 | 456 | 375 | 387 | 400 | 387 | 350 | 406 | 393 | 387 | 387 | 400 | 418 | 387 | 381 | 375 | 368 | 362 | 318 | 393 | 350 |
| 400 | 418 | 368 | 362 | 0 | 0 | 0 | 0 | 0 | 375 | 418 | 381 | 456 | 431 | 375 | 418 | 418 | 375 | 431 | 431 | 362 | 393 | 406 | 368 | 387 | 387 | 362 | 375 | 362 | 356 | 0 | 318 |
| 356 | 350 | 331 | 0 | 0 | 0 | 0 | 0 | 0 | 325 | 325 | 375 | 362 | 350 | 368 | 350 | 343 | 350 | 331 | 325 | 331 | 350 | 312 | 356 | 0 | 0 | 318 | 0 | 325 | 0 | 0 | 0 |
| 325 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 312 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 325 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 312 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 343 | 375 | 331 | 312 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 393 | 343 | 331 | 325 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 418 | 362 | 331 | 331 | 331 | 312 | 312 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 387 | 375 | 400 | 375 | 375 | 375 | 356 | 318 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 400 | 418 | 431 | 381 | 468 | 431 | 393 | 443 | 356 | 312 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 462 | 400 | 418 | 406 | 418 | 406 | 456 | 406 | 331 | 325 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 418 | 456 | 443 | 406 | 462 | 462 | 443 | 443 | 418 | 362 | 331 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 493 | 481 | 406 | 462 | 512 | 468 | 493 | 468 | 418 | 400 | 362 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 531 | 531 | 568 | 543 | 531 | 549 | 543 | 468 | 468 | 381 | 343 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 531 | 543 | 493 | 481 | 531 | 468 | 512 | 531 | 393 | 381 | 325 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 462 | 462 | 393 | 443 | 443 | 387 | 418 | 393 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 418 | 443 | 456 | 418 | 418 | 375 | 387 | 387 | 312 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 393 | 368 | 325 | 381 | 331 | 0 | 362 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

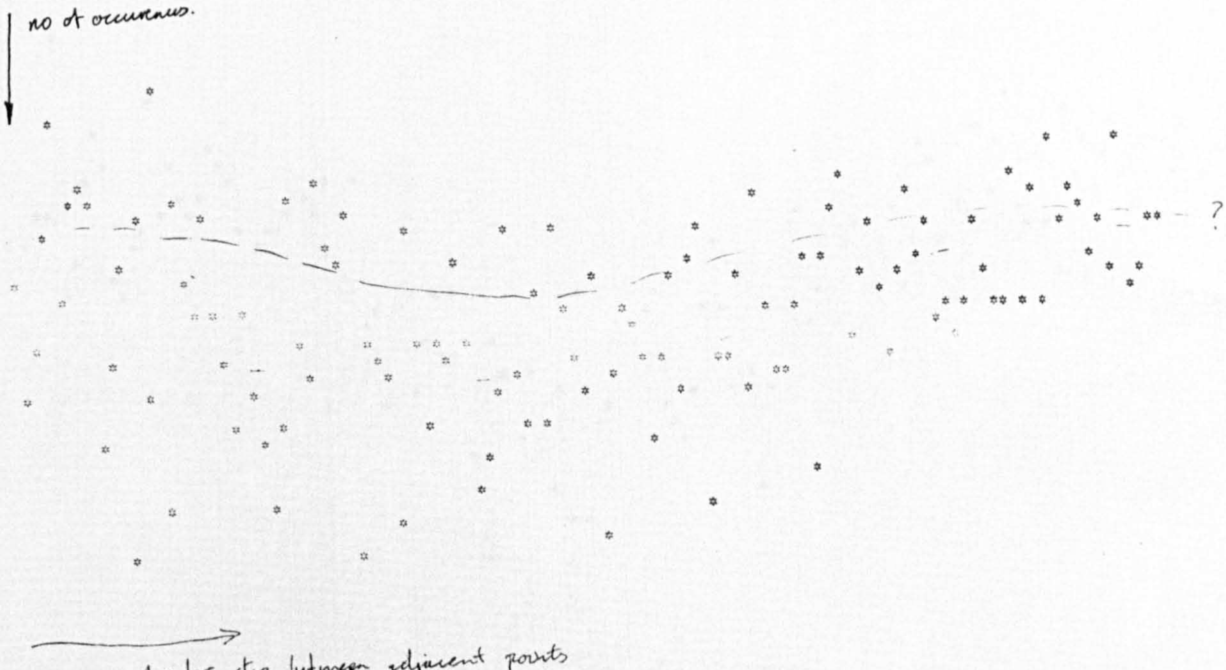
**5 b) Thresholded Jeep image from Alabama database (right half)**




6 a). Phase plane step histogram for 8 by 8 square test object.

.LOG:6
JUMP EXTRACTOR

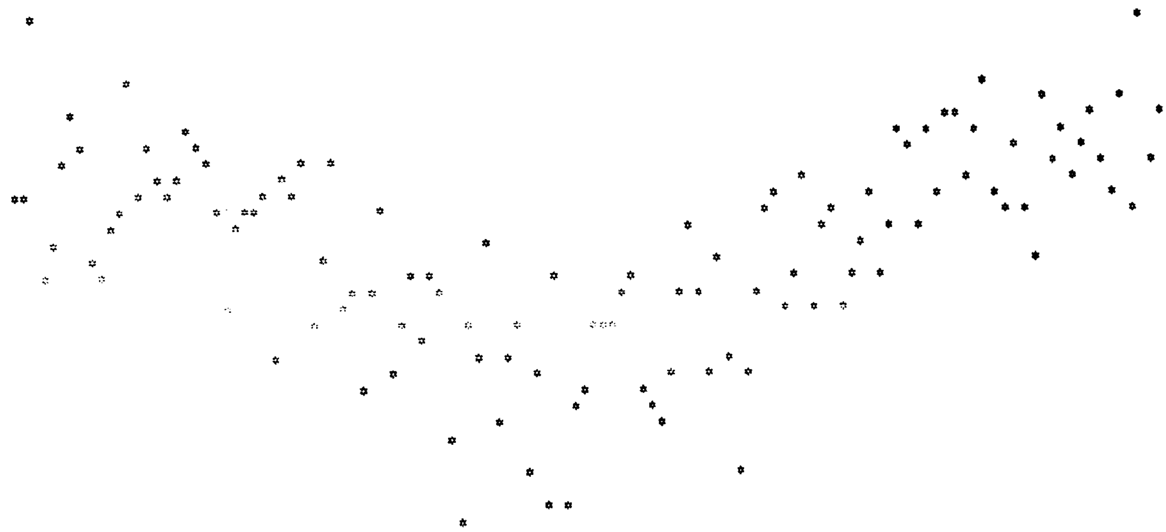
25-JAN-1984 16:10 Page 14




ROTATION REMOVED x direction : 298.1250 Y direction : 0.0000

6 b). Phase plane step histogram for Jeep from Alabama database

:7 EXTRACTOR 25-JAN-1984 16:13 Page 14



ROTATION REMOVED X direction : 140.6250 Y direction : 5.6250

**6 c). Phase plane step histogram for thresholded jeep from Alabama database**

FT.LOG;1 4-JAN-1984 11:21 Page 6

**** INPUT DATA ****

INPUT FILES REAL : PIO4TGTK IMAGINARY : NULL0000

REAL PART LEFT HALF SCALE FACTOR = 7.8047

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 523 | 492 | 507 | 500 | 476 | 523 | 484 | 492 | 484 | 460 | 484 | 484 | 492 | 507 | 507 | 460 | 507 | 554 | 507 | 507 | 570 | 554 | 523 | 492 | 476 | 468 | 484 | 523 | 523 | 507 | 507 | 484 |
| 492 | 500 | 500 | 484 | 500 | 500 | 476 | 453 | 468 | 476 | 476 | 500 | 484 | 500 | 523 | 484 | 500 | 523 | 554 | 500 | 523 | 523 | 507 | 500 | 484 | 476 | 460 | 492 | 484 | 500 | 484 | 492 |
| 523 | 484 | 476 | 468 | 460 | 453 | 398 | 492 | 437 | 476 | 437 | 445 | 484 | 468 | 484 | 476 | 484 | 492 | 500 | 507 | 507 | 539 | 484 | 492 | 492 | 484 | 476 | 484 | 484 | 484 | 468 | 460 |
| 460 | 434 | 484 | 453 | 468 | 453 | 445 | 382 | 398 | 445 | 429 | 460 | 523 | 500 | 507 | 601 | 523 | 500 | 640 | 570 | 539 | 601 | 523 | 523 | 539 | 484 | 484 | 539 | 500 | 484 | 523 | 492 |
| 445 | 375 | 375 | 398 | 375 | 406 | 375 | 351 | 375 | 375 | 375 | 375 | 437 | 437 | 460 | 460 | 476 | 468 | 523 | 460 | 484 | 500 | 484 | 460 | 476 | 460 | 468 | 492 | 460 | 484 | 468 | 429 |
| 265 | 273 | 273 | 289 | 289 | 312 | 312 | 312 | 273 | 289 | 343 | 375 | 406 | 437 | 468 | 468 | 476 | 468 | 492 | 492 | 453 | 476 | 492 | 437 | 476 | 460 | 460 | 468 | 460 | 437 | 468 | 437 |
| 312 | 265 | 250 | 289 | 312 | 250 | 320 | 312 | 250 | 320 | 336 | 273 | 406 | 414 | 406 | 484 | 484 | 453 | 523 | 468 | 453 | 492 | 484 | 414 | 460 | 484 | 484 | 468 | 468 | 476 | 460 | 453 |
| 211 | 219 | 219 | 250 | 250 | 258 | 312 | 289 | 273 | 336 | 312 | 375 | 375 | 398 | 429 | 453 | 476 | 484 | 484 | 460 | 484 | 460 | 445 | 476 | 453 | 437 | 437 | 437 | 460 | 437 | 437 | 406 |
| 281 | 211 | 187 | 312 | 219 | 258 | 375 | 281 | 250 | 351 | 312 | 281 | 468 | 382 | 375 | 476 | 453 | 390 | 507 | 429 | 406 | 484 | 484 | 414 | 468 | 445 | 390 | 453 | 445 | 406 | 453 | 445 |
| 219 | 180 | 250 | 234 | 234 | 281 | 281 | 265 | 289 | 273 | 312 | 351 | 320 | 320 | 375 | 320 | 320 | 406 | 375 | 336 | 445 | 336 | 351 | 460 | 320 | 375 | 398 | 351 | 398 | 414 | 398 | 390 |
| 180 | 180 | 203 | 211 | 211 | 250 | 250 | 289 | 250 | 250 | 265 | 265 | 265 | 281 | 273 | 281 | 289 | 281 | 289 | 273 | 273 | 273 | 312 | 273 | 312 | 375 | 320 | 312 | 343 | 320 | 375 | 406 |
| 156 | 164 | 187 | 172 | 211 | 203 | 195 | 211 | 219 | 211 | 250 | 250 | 250 | 211 | 211 | 180 | 211 | 219 | 187 | 219 | 320 | 211 | 211 | 250 | 211 | 250 | 258 | 250 | 250 | 265 | 265 | 258 |
| 148 | 156 | 148 | 156 | 187 | 195 | 211 | 219 | 211 | 219 | 203 | 211 | 250 | 211 | 211 | 203 | 211 | 211 | 211 | 180 | 187 | 211 | 187 | 195 | 265 | 211 | 211 | 258 | 211 | 211 | 258 | 281 |
| 195 | 164 | 164 | 203 | 172 | 156 | 211 | 195 | 250 | 211 | 219 | 211 | 234 | 211 | 195 | 211 | 234 | 180 | 195 | 164 | 164 | 195 | 180 | 180 | 195 | 172 | 180 | 211 | 172 | 172 | 211 | 187 |
| 148 | 148 | 148 | 125 | 148 | 156 | 148 | 156 | 164 | 156 | 156 | 164 | 180 | 156 | 164 | 180 | 164 | 180 | 148 | 148 | 164 | 148 | 125 | 125 | 148 | 125 | 125 | 164 | 148 | 133 | 148 | 164 |
| 156 | 156 | 180 | 133 | 164 | 156 | 180 | 156 | 156 | 156 | 148 | 180 | 164 | 164 | 180 | 187 | 164 | 187 | 219 | 180 | 187 | 203 | 180 | 164 | 187 | 156 | 164 | 187 | 156 | 156 | 164 | 156 |
| 125 | 148 | 101 | 101 | 164 | 125 | 125 | 156 | 125 | 148 | 117 | 125 | 133 | 156 | 156 | 164 | 164 | 187 | 156 | 180 | 203 | 148 | 172 | 187 | 148 | 156 | 156 | 148 | 156 | 156 | 125 | 101 |
| 148 | 148 | 125 | 101 | 148 | 148 | 148 | 148 | 125 | 125 | 148 | 148 | 125 | 125 | 156 | 164 | 180 | 211 | 187 | 219 | 211 | 203 | 195 | 187 | 172 | 180 | 172 | 156 | 180 | 148 | 156 | 101 |
| 187 | 156 | 148 | 164 | 148 | 125 | 187 | 164 | 125 | 164 | 156 | 172 | 156 | 148 | 172 | 164 | 187 | 211 | 187 | 265 | 234 | 211 | 203 | 211 | 203 | 195 | 172 | 148 | 156 | 148 | 101 | 172 |
| 180 | 187 | 180 | 156 | 211 | 187 | 164 | 172 | 219 | 164 | 180 | 211 | 172 | 195 | 211 | 195 | 211 | 211 | 250 | 289 | 250 | 273 | 258 | 211 | 258 | 265 | 265 | 211 | 211 | 250 | 203 | 180 |
| 164 | 148 | 125 | 148 | 117 | 125 | 148 | 148 | 117 | 148 | 133 | 101 | 125 | 164 | 156 | 180 | 187 | 211 | 250 | 273 | 273 | 250 | 273 | 250 | 250 | 250 | 180 | 250 | 187 | 125 | 164 | 180 |
| 148 | 148 | 156 | 148 | 148 | 187 | 101 | 164 | 148 | 156 | 148 | 148 | 148 | 156 | 156 | 164 | 180 | 250 | 250 | 265 | 281 | 273 | 265 | 273 | 312 | 273 | 265 | 273 | 258 | 219 | 203 | 211 |
| 156 | 156 | 172 | 180 | 156 | 156 | 164 | 164 | 164 | 148 | 148 | 125 | 164 | 148 | 164 | 180 | 164 | 219 | 250 | 265 | 289 | 320 | 289 | 312 | 336 | 265 | 312 | 336 | 273 | 265 | 250 | 250 |
| 195 | 164 | 125 | 172 | 172 | 164 | 164 | 156 | 125 | 164 | 172 | 148 | 164 | 172 | 148 | 211 | 180 | 211 | 250 | 312 | 312 | 320 | 351 | 320 | 375 | 398 | 406 | 398 | 375 | 375 | 336 | 343 |
| 125 | 164 | 156 | 148 | 148 | 156 | 156 | 148 | 156 | 125 | 125 | 164 | 148 | 125 | 164 | 156 | 156 | 180 | 234 | 250 | 265 | 312 | 312 | 343 | 375 | 375 | 390 | 390 | 375 | 351 | 320 | 336 |
| 86 | 148 | 156 | 117 | 125 | 125 | 86 | 125 | 133 | 78 | 86 | 125 | 70 | 86 | 125 | 117 | 125 | 164 | 180 | 180 | 250 | 289 | 289 | 375 | 382 | 343 | 390 | 406 | 398 | 375 | 414 | 398 |
| 148 | 164 | 164 | 125 | 148 | 180 | 125 | 86 | 187 | 117 | 125 | 180 | 125 | 86 | 187 | 117 | 94 | 250 | 187 | 156 | 336 | 273 | 250 | 429 | 375 | 336 | 414 | 414 | 336 | 429 | 398 | 320 |
| 125 | 148 | 125 | 148 | 180 | 125 | 148 | 101 | 125 | 148 | 86 | 125 | 125 | 94 | 101 | 148 | 101 | 180 | 203 | 164 | 258 | 312 | 250 | 375 | 375 | 351 | 398 | 476 | 414 | 453 | 476 | 406 |
| 180 | 164 | 180 | 148 | 180 | 156 | 148 | 164 | 180 | 86 | 164 | 148 | 164 | 117 | 156 | 148 | 172 | 180 | 180 | 203 | 250 | 273 | 281 | 343 | 375 | 406 | 453 | 460 | 500 | 601 | 492 | 484 |
| 195 | 211 | 164 | 211 | 211 | 187 | 195 | 211 | 187 | 211 | 180 | 172 | 187 | 180 | 211 | 180 | 187 | 195 | 203 | 211 | 250 | 250 | 281 | 375 | 398 | 406 | 468 | 663 | 734 | 710 | 710 | 663 |
| 164 | 101 | 180 | 164 | 148 | 203 | 180 | 164 | 180 | 187 | 180 | 180 | 156 | 148 | 180 | 180 | 164 | 180 | 180 | 211 | 211 | 234 | 265 | 312 | 351 | 406 | 414 | 617 | 710 | 718 | 687 | 663 |
| 187 | 195 | 180 | 203 | 211 | 211 | 219 | 211 | 187 | 211 | 187 | 187 | 203 | 164 | 187 | 180 | 180 | 172 | 164 | 172 | 180 | 211 | 211 | 320 | 336 | 390 | 460 | 734 | 812 | 843 | 859 | 780 |
| 164 | 180 | 211 | 187 | 203 | 219 | 211 | 211 | 211 | 211 | 211 | 187 | 164 | 180 | 172 | 180 | 172 | 148 | 164 | 172 | 164 | 211 | 211 | 265 | 312 | 398 | 445 | 710 | 804 | 882 | 882 | 804 |
| 211 | 273 | 250 | 250 | 281 | 234 | 250 | 250 | 250 | 250 | 250 | 211 | 203 | 203 | 211 | 180 | 195 | 203 | 180 | 187 | 187 | 203 | 234 | 258 | 289 | 390 | 484 | 788 | 882 | 999 | 890 | 843 |
| 195 | 203 | 219 | 195 | 195 | 211 | 219 | 250 | 195 | 211 | 211 | 187 | 187 | 203 | 164 | 180 | 180 | 125 | 156 | 156 | 164 | 180 | 187 | 211 | 312 | 336 | 390 | 663 | 812 | 874 | 890 | 859 |
| 250 | 211 | 211 | 219 | 180 | 211 | 242 | 211 | 187 | 219 | 203 | 164 | 211 | 211 | 203 | 187 | 180 | 172 | 211 | 187 | 187 | 219 | 211 | 242 | 273 | 312 | 375 | 507 | 617 | 718 | 679 | 617 |
| 187 | 203 | 265 | 203 | 211 | 250 | 211 | 187 | 211 | 234 | 156 | 203 | 172 | 164 | 187 | 164 | 125 | 172 | 156 | 125 | 187 | 164 | 164 | 250 | 273 | 250 | 375 | 507 | 578 | 679 | 710 | 578 |
| 211 | 164 | 180 | 187 | 250 | 172 | 187 | 180 | 164 | 211 | 172 | 187 | 187 | 203 | 180 | 180 | 164 | 180 | 180 | 156 | 164 | 180 | 172 | 211 | 234 | 250 | 289 | 390 | 460 | 484 | 468 | 460 |
| 211 | 187 | 187 | 172 | 180 | 180 | 180 | 187 | 180 | 211 | 180 | 180 | 187 | 180 | 180 | 180 | 180 | 164 | 172 | 187 | 180 | 180 | 180 | 211 | 219 | 258 | 273 | 437 | 382 | 390 | 375 | 390 |
| 203 | 211 | 203 | 195 | 203 | 219 | 187 | 187 | 211 | 195 | 234 | 250 | 250 | 203 | 219 | 195 | 180 | 211 | 211 | 211 | 211 | 250 | 211 | 250 | 250 | 250 | 273 | 273 | 320 | 320 | 289 | 281 |
| 211 | 187 | 234 | 203 | 195 | 219 | 180 | 164 | 250 | 211 | 187 | 234 | 187 | 187 | 250 | 156 | 180 | 211 | 180 | 195 | 265 | 180 | 211 | 265 | 180 | 234 | 320 | 234 | 250 | 375 | 234 | 273 |
| 187 | 180 | 211 | 180 | 250 | 187 | 164 | 172 | 187 | 180 | 187 | 180 | 187 | 203 | 203 | 172 | 211 | 250 | 187 | 250 | 211 | 180 | 219 | 211 | 234 | 234 | 234 | 211 | 258 | 265 | 234 | 273 |
| 211 | 195 | 187 | 211 | 180 | 203 | 203 | 203 | 203 | 211 | 211 | 195 | 211 | 234 | 219 | 250 | 250 | 211 | 250 | 273 | 211 | 219 | 234 | 250 | 211 | 250 | 250 | 234 | 265 | 281 | 273 | 250 |
| 234 | 258 | 258 | 250 | 250 | 250 | 265 | 250 | 258 | 250 | 258 | 265 | 265 | 312 | 273 | 258 | 312 | 312 | 250 | 312 | 289 | 281 | 273 | 320 | 320 | 312 | 281 | 312 | 320 | 320 | 343 | 375 |
| 211 | 211 | 211 | 250 | 219 | 219 | 250 | 250 | 250 | 250 | 250 | 211 | 250 | 250 | 234 | 250 | 250 | 250 | 265 | 250 | 273 | 234 | 250 | 281 | 273 | 289 | 320 | 289 | 289 | 351 | 343 | 320 |
| 281 | 250 | 234 | 273 | 250 | 250 | 289 | 289 | 234 | 273 | 273 | 219 | 312 | 273 | 258 | 312 | 273 | 281 | 320 | 289 | 250 | 312 | 281 | 273 | 312 | 320 | 320 | 312 | 375 | 382 | 375 | 406 |
| 289 | 281 | 273 | 289 | 312 | 258 | 289 | 375 | 250 | 265 | 320 | 265 | 273 | 343 | 273 | 336 | 375 | 289 | 320 | 390 | 320 | 320 | 375 | 320 | 312 | 390 | 375 | 320 | 429 | 375 | 390 | 476 |
| 336 | 273 | 265 | 320 | 273 | 265 | 343 | 406 | 265 | 320 | 375 | 250 | 289 | 375 | 250 | 351 | 406 | 289 | 375 | 437 | 281 | 375 | 390 | 273 | 320 | 375 | 312 | 351 | 375 | 375 | 414 | 484 |
| 336 | 375 | 312 | 343 | 382 | 343 | 375 | 351 | 320 | 312 | 312 | 312 | 320 | 320 | 320 | 375 | 375 | 375 | 289 | 375 | 375 | 343 | 375 | 351 | 375 | 312 | 312 | 281 | 343 | 375 | 429 | 453 |
| 460 | 429 | 375 | 476 | 429 | 445 | 476 | 460 | 437 | 476 | 429 | 429 | 445 | 414 | 398 | 453 | 414 | 414 | 476 | 437 | 453 | 476 | 437 | 382 | 398 | 375 | 390 | 414 | 390 | 429 | 476 | 484 |
| 406 | 492 | 476 | 398 | 500 | 460 | 398 | 476 | 492 | 382 | 437 | 476 | 375 | 437 | 414 | 375 | 453 | 453 | 320 | 437 | 445 | 406 | 429 | 406 | 398 | 390 | 375 | 375 | 437 | 406 | 437 | 500 |
| 406 | 460 | 484 | 406 | 460 | 434 | 414 | 539 | 453 | 414 | 507 | 429 | 406 | 523 | 437 | 406 | 484 | 429 | 460 | 500 | 414 | 460 | 484 | 375 | 390 | 414 | 375 | 406 | 468 | 414 | 437 | 507 |
| 468 | 523 | 507 | 492 | 507 | 523 | 484 | 476 | 507 | 492 | 320 | 507 | 484 | 453 | 476 | 484 | 437 | 484 | 460 | 468 | 484 | 445 | 453 | 460 | 453 | 406 | 429 | 390 | 453 | 437 | 453 | 484 |
| 578 | 535 | 535 | 585 | 578 | 535 | 535 | 554 | 585 | 523 | 539 | 578 | 523 | 500 | 554 | 507 | 539 | 523 | 492 | 554 | 554 | 570 | 507 | 523 | 500 | 484 | 468 | 523 | 523 | 484 | 539 | 539 |
| 539 | 617 | 570 | 500 | 570 | 570 | 500 | 523 | 523 | 484 | 492 | 554 | 484 | 484 | 492 | 484 | 484 | 484 | 500 | 445 | 507 | 437 | 476 | 492 | 453 | 453 | 570 | 460 | 468 | 507 | 460 | 484 |

7 a). Detail of tank object from Alabama database (left hlaf)

FT.LOG:1 4-JAN-1984 11:21 Page 7

**** INPUT DATA ****

INPUT FILES REAL : PI04TGTK IMAGINARY : NULL0000

REAL PART RIGHT HALF SCALE FACTOR = 7.8047

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 492 | 523 | 484 | 476 | 492 | 500 | 492 | 523 | 554 | 523 | 539 | 523 | 507 | 484 | 500 | 484 | 507 | 507 | 507 | 476 | 523 | 492 | 492 | 539 | 492 | 523 | 507 | 523 | 523 | 523 | 507 | 500 |
| 484 | 507 | 507 | 476 | 484 | 507 | 484 | 523 | 523 | 500 | 523 | 500 | 484 | 492 | 492 | 484 | 492 | 523 | 484 | 507 | 484 | 500 | 484 | 523 | 484 | 539 | 507 | 523 | 484 | 507 | 484 | 500 |
| 476 | 492 | 476 | 468 | 500 | 453 | 484 | 539 | 484 | 507 | 484 | 476 | 484 | 476 | 484 | 507 | 484 | 468 | 492 | 460 | 484 | 476 | 492 | 484 | 492 | 484 | 500 | 484 | 484 | 585 | 492 | 507 |
| 500 | 523 | 484 | 484 | 578 | 539 | 484 | 601 | 539 | 523 | 523 | 523 | 484 | 484 | 523 | 492 | 492 | 523 | 492 | 523 | 507 | 476 | 500 | 523 | 523 | 507 | 554 | 484 | 484 | 570 | 500 | 500 |
| 484 | 468 | 445 | 484 | 469 | 445 | 484 | 484 | 484 | 523 | 492 | 468 | 484 | 484 | 460 | 484 | 484 | 476 | 468 | 476 | 476 | 484 | 468 | 468 | 484 | 484 | 484 | 492 | 500 | 484 | 468 | 507 |
| 468 | 414 | 390 | 437 | 414 | 468 | 468 | 468 | 453 | 460 | 468 | 468 | 468 | 460 | 460 | 468 | 476 | 414 | 453 | 468 | 468 | 437 | 468 | 437 | 460 | 476 | 460 | 460 | 453 | 414 | 468 | 468 |
| 406 | 429 | 445 | 445 | 476 | 460 | 453 | 484 | 492 | 484 | 437 | 484 | 460 | 414 | 476 | 453 | 398 | 460 | 484 | 406 | 484 | 468 | 390 | 476 | 460 | 414 | 460 | 445 | 445 | 453 | 468 | 492 |
| 437 | 390 | 429 | 445 | 398 | 445 | 437 | 445 | 453 | 437 | 445 | 460 | 437 | 445 | 468 | 437 | 445 | 468 | 453 | 453 | 437 | 484 | 453 | 406 | 484 | 414 | 382 | 437 | 445 | 429 | 453 | 453 |
| 375 | 406 | 406 | 375 | 406 | 398 | 375 | 414 | 429 | 390 | 390 | 468 | 390 | 375 | 437 | 398 | 375 | 437 | 406 | 406 | 429 | 390 | 390 | 414 | 406 | 390 | 406 | 390 | 375 | 390 | 390 | 382 |
| 437 | 375 | 343 | 406 | 320 | 320 | 398 | 312 | 375 | 398 | 429 | 375 | 445 | 414 | 375 | 437 | 429 | 382 | 445 | 445 | 375 | 437 | 414 | 336 | 453 | 390 | 336 | 429 | 375 | 336 | 445 | 406 |
| 351 | 351 | 320 | 289 | 289 | 281 | 250 | 289 | 312 | 250 | 258 | 351 | 234 | 281 | 375 | 289 | 273 | 351 | 273 | 273 | 336 | 281 | 265 | 312 | 281 | 273 | 320 | 289 | 265 | 281 | 312 | 273 |
| 312 | 265 | 281 | 250 | 211 | 203 | 250 | 187 | 203 | 312 | 250 | 211 | 312 | 320 | 250 | 281 | 320 | 265 | 258 | 265 | 250 | 273 | 265 | 250 | 250 | 219 | 211 | 273 | 250 | 211 | 219 | 250 |
| 211 | 312 | 273 | 180 | 289 | 211 | 180 | 211 | 203 | 180 | 180 | 250 | 187 | 195 | 211 | 211 | 219 | 211 | 211 | 195 | 265 | 187 | 211 | 211 | 180 | 187 | 234 | 156 | 211 | 219 | 180 | 195 |
| 195 | 234 | 211 | 180 | 180 | 180 | 195 | 195 | 164 | 211 | 164 | 172 | 195 | 148 | 172 | 187 | 156 | 172 | 180 | 156 | 164 | 172 | 172 | 172 | 164 | 148 | 172 | 164 | 164 | 172 | 211 | 187 |
| 125 | 125 | 148 | 149 | 117 | 148 | 125 | 101 | 148 | 101 | 125 | 94 | 148 | 125 | 125 | 148 | 117 | 117 | 125 | 125 | 94 | 148 | 117 | 101 | 101 | 117 | 156 | 86 | 156 | 148 | 117 | 148 |
| 148 | 180 | 156 | 125 | 148 | 133 | 133 | 164 | 125 | 148 | 101 | 125 | 101 | 101 | 133 | 101 | 125 | 125 | 156 | 125 | 133 | 86 | 125 | 125 | 86 | 86 | 164 | 101 | 125 | 164 | 101 | 133 |
| 156 | 117 | 94 | 125 | 94 | 125 | 164 | 101 | 117 | 94 | 180 | 78 | 125 | 172 | 78 | 148 | 172 | 78 | 101 | 172 | 39 | 148 | 187 | 70 | 125 | 172 | 94 | 156 | 180 | 101 | 148 | 187 |
| 125 | 117 | 125 | 117 | 117 | 94 | 125 | 101 | 78 | 125 | 117 | 62 | 125 | 86 | 70 | 125 | 101 | 70 | 125 | 117 | 16 | 125 | 101 | 39 | 125 | 94 | 94 | 125 | 125 | 101 | 101 | 125 |
| 149 | 94 | 148 | 101 | 117 | 148 | 101 | 101 | 94 | 148 | 94 | 125 | 94 | 101 | 78 | 125 | 86 | 101 | 101 | 117 | 125 | 101 | 101 | 125 | 86 | 101 | 101 | 94 | 148 | 94 | 94 | 94 |
| 250 | 164 | 180 | 211 | 187 | 164 | 187 | 164 | 164 | 187 | 148 | 117 | 203 | 164 | 148 | 172 | 164 | 125 | 164 | 125 | 148 | 164 | 180 | 148 | 156 | 117 | 125 | 148 | 148 | 125 | 164 | 148 |
| 70 | 148 | 164 | 78 | 147 | 133 | 39 | 125 | 125 | 47 | 125 | 125 | 47 | 125 | 125 | 86 | 101 | 125 | 62 | 117 | 94 | 39 | 117 | 86 | 70 | 180 | 125 | 23 | 156 | 117 | 78 | 164 |
| 164 | 137 | 137 | 187 | 156 | 148 | 148 | 125 | 101 | 78 | 125 | 117 | 164 | 148 | 164 | 156 | 164 | 164 | 148 | 148 | 117 | 148 | 148 | 148 | 125 | 156 | 180 | 156 | 156 | 164 | 148 | 86 |
| 250 | 211 | 211 | 211 | 156 | 187 | 172 | 156 | 164 | 125 | 148 | 148 | 156 | 125 | 172 | 156 | 172 | 164 | 172 | 172 | 125 | 156 | 164 | 156 | 148 | 180 | 164 | 180 | 187 | 164 | 148 | 156 |
| 289 | 312 | 312 | 312 | 250 | 250 | 273 | 211 | 211 | 211 | 195 | 211 | 234 | 211 | 250 | 273 | 211 | 250 | 258 | 211 | 250 | 234 | 203 | 211 | 250 | 219 | 250 | 258 | 211 | 211 | 211 | 211 |
| 312 | 289 | 312 | 320 | 250 | 211 | 211 | 203 | 203 | 180 | 195 | 172 | 164 | 187 | 187 | 195 | 211 | 211 | 187 | 187 | 195 | 164 | 195 | 211 | 211 | 211 | 250 | 203 | 211 | 219 | 203 | 203 |
| 382 | 343 | 343 | 336 | 289 | 250 | 250 | 242 | 203 | 203 | 180 | 172 | 180 | 180 | 187 | 203 | 211 | 211 | 203 | 211 | 211 | 172 | 187 | 203 | 203 | 211 | 250 | 211 | 234 | 250 | 195 | 219 |
| 406 | 351 | 312 | 375 | 312 | 211 | 273 | 234 | 180 | 187 | 250 | 164 | 211 | 211 | 180 | 219 | 219 | 211 | 234 | 234 | 187 | 211 | 211 | 164 | 211 | 265 | 219 | 258 | 273 | 258 | 250 | 273 |
| 453 | 429 | 343 | 375 | 312 | 250 | 320 | 265 | 219 | 234 | 219 | 172 | 180 | 211 | 203 | 234 | 250 | 250 | 219 | 219 | 250 | 203 | 219 | 203 | 195 | 234 | 250 | 250 | 250 | 281 | 250 | 265 |
| 523 | 460 | 406 | 437 | 336 | 336 | 375 | 265 | 258 | 312 | 273 | 211 | 250 | 234 | 219 | 250 | 234 | 211 | 250 | 250 | 211 | 273 | 250 | 211 | 219 | 211 | 211 | 258 | 265 | 265 | 289 | 265 |
| 523 | 500 | 468 | 427 | 406 | 390 | 351 | 320 | 343 | 312 | 312 | 273 | 250 | 273 | 234 | 234 | 265 | 273 | 250 | 273 | 320 | 281 | 273 | 312 | 250 | 265 | 281 | 258 | 265 | 273 | 312 | 273 |
| 539 | 492 | 453 | 414 | 398 | 382 | 375 | 375 | 312 | 320 | 375 | 273 | 351 | 320 | 250 | 312 | 289 | 250 | 312 | 351 | 265 | 375 | 343 | 250 | 320 | 336 | 250 | 320 | 320 | 250 | 351 | 320 |
| 663 | 507 | 453 | 484 | 406 | 406 | 390 | 375 | 336 | 343 | 375 | 320 | 312 | 312 | 258 | 281 | 312 | 273 | 273 | 320 | 320 | 312 | 343 | 336 | 312 | 312 | 273 | 312 | 343 | 250 | 343 | 312 |
| 710 | 523 | 500 | 460 | 429 | 429 | 406 | 390 | 375 | 375 | 375 | 336 | 336 | 320 | 281 | 312 | 312 | 289 | 312 | 289 | 320 | 375 | 375 | 320 | 375 | 351 | 320 | 320 | 343 | 312 | 312 | 289 |
| 812 | 578 | 507 | 539 | 484 | 484 | 460 | 468 | 445 | 453 | 437 | 437 | 406 | 406 | 398 | 392 | 398 | 375 | 375 | 414 | 406 | 406 | 406 | 414 | 429 | 429 | 375 | 390 | 382 | 375 | 375 | 375 |
| 663 | 539 | 523 | 453 | 476 | 492 | 460 | 437 | 453 | 437 | 406 | 390 | 406 | 375 | 406 | 351 | 320 | 375 | 375 | 312 | 390 | 343 | 382 | 406 | 375 | 375 | 390 | 375 | 336 | 351 | 375 | 320 |
| 617 | 453 | 484 | 492 | 484 | 492 | 476 | 507 | 484 | 468 | 468 | 453 | 507 | 460 | 429 | 460 | 414 | 414 | 390 | 343 | 406 | 390 | 398 | 437 | 414 | 406 | 406 | 460 | 390 | 437 | 414 | 359 |
| 507 | 484 | 398 | 476 | 468 | 453 | 484 | 468 | 453 | 453 | 453 | 437 | 429 | 437 | 476 | 437 | 390 | 414 | 382 | 375 | 414 | 336 | 375 | 429 | 390 | 390 | 390 | 406 | 375 | 398 | 375 | 336 |
| 414 | 406 | 453 | 406 | 453 | 523 | 453 | 492 | 492 | 453 | 460 | 484 | 453 | 414 | 500 | 414 | 414 | 453 | 390 | 336 | 453 | 375 | 343 | 453 | 406 | 375 | 429 | 406 | 375 | 414 | 390 | 320 |
| 343 | 375 | 375 | 382 | 437 | 414 | 484 | 437 | 468 | 468 | 484 | 445 | 484 | 437 | 453 | 445 | 414 | 406 | 414 | 375 | 375 | 406 | 429 | 414 | 437 | 414 | 414 | 414 | 406 | 406 | 406 | 429 |
| 312 | 312 | 320 | 375 | 406 | 453 | 460 | 507 | 453 | 507 | 484 | 492 | 476 | 484 | 476 | 484 | 484 | 460 | 468 | 437 | 445 | 437 | 460 | 437 | 476 | 437 | 468 | 476 | 476 | 476 | 460 | 460 |
| 351 | 250 | 273 | 375 | 336 | [illegible] | 484 | 406 | 484 | 539 | 476 | 500 | 554 | 484 | 460 | 484 | 460 | 460 | 484 | 460 | 437 | 484 | 484 | 453 | 492 | 484 | 414 | 523 | 492 | 414 | 523 | 468 |
| 250 | 219 | 273 | 273 | 312 | 375 | 484 | 414 | 437 | 484 | 507 | 453 | 468 | 476 | 484 | 484 | 460 | 484 | 437 | 437 | 492 | 453 | 453 | 484 | 460 | 453 | 429 | 507 | 468 | 453 | 492 | 460 |
| 250 | 231 | 312 | 289 | 375 | 406 | 414 | 468 | 484 | 484 | 484 | 476 | 468 | 460 | 484 | 484 | 437 | 476 | 453 | 445 | 476 | 460 | 453 | 484 | 468 | 484 | 484 | 476 | 492 | 492 | 460 | 492 |
| 351 | 351 | 390 | 390 | 406 | 453 | 460 | 500 | 523 | 507 | 539 | 585 | 539 | 578 | 523 | 507 | 523 | 500 | 523 | 523 | 500 | 523 | 590 | 507 | 539 | 554 | 507 | 585 | 585 | 500 | 523 | 570 |
| 382 | 375 | 375 | 414 | 398 | 429 | 437 | 468 | 453 | 492 | 484 | 484 | 492 | 454 | 476 | 539 | 507 | 484 | 539 | 484 | 468 | 539 | 484 | 476 | 578 | 523 | 476 | 554 | 523 | 484 | 539 | 507 |
| 414 | 437 | 476 | 476 | 460 | 492 | 492 | 492 | 523 | 500 | 476 | 523 | 523 | 484 | 523 | 523 | 484 | 523 | 492 | 476 | 523 | 523 | 476 | 484 | 523 | 484 | 539 | 539 | 500 | 523 | 539 | 484 |
| 429 | 476 | 523 | 484 | 484 | 554 | 507 | 507 | 523 | 484 | 500 | 523 | 492 | 492 | 523 | 523 | 492 | 578 | 523 | 507 | 523 | 523 | 523 | 585 | 539 | 492 | 523 | 570 | 554 | 578 | 492 | 507 |
| 437 | 492 | 578 | 476 | 507 | 578 | 484 | 507 | 585 | 492 | 437 | 539 | 492 | 437 | 578 | 523 | 460 | 585 | 523 | 476 | 570 | 554 | 484 | 601 | 578 | 484 | 554 | 523 | 523 | 578 | 523 | 484 |
| 460 | 492 | 484 | 492 | 484 | [illegible] | 500 | [illegible] | 507 | 500 | 492 | 523 | 492 | 523 | 554 | 492 | 554 | 523 | 523 | 570 | 554 | 570 | 507 | 585 | 523 | 523 | 585 | 539 | 507 | 554 | 492 | 484 |
| 492 | 523 | 570 | [illegible] | [illegible] | [illegible] | 507 | [illegible] | 523 | 570 | 574 | 554 | 565 | 554 | 579 | 578 | 617 | 585 | 601 | 585 | 585 | 617 | 539 | 640 | 640 | 535 | 570 | 500 | 617 | 578 | 585 | 570 |
| 484 | 507 | [illegible] | [illegible] | [illegible] | [illegible] | 554 | [illegible] | [illegible] | 507 | 432 | 578 | 539 | 523 | 585 | 585 | 523 | 617 | 554 | 539 | 663 | 554 | 570 | 579 | 539 | 570 | 663 | 523 | 570 | 679 | 523 | 570 |
| 492 | 484 | 570 | 523 | 523 | [illegible] | 507 | [illegible] | 554 | 507 | 523 | 595 | 492 | 539 | 601 | 539 | 554 | 585 | 539 | 554 | 570 | 507 | 539 | 578 | 523 | 585 | 578 | 500 | 492 | 617 | 453 | 523 |
| 476 | 523 | 554 | 523 | 523 | 554 | 539 | 507 | 554 | 492 | 523 | 507 | 523 | 578 | 570 | 539 | 578 | 554 | 523 | 585 | 578 | 523 | 535 | 570 | 523 | 554 | 484 | 523 | 523 | 523 | 539 | 523 |
| 492 | 570 | 601 | 554 | 595 | 570 | 539 | 595 | 640 | 570 | 578 | 585 | 517 | 601 | 663 | 640 | 640 | 640 | 578 | 663 | 640 | 578 | 663 | 663 | 578 | 601 | 570 | 507 | 601 | 601 | 554 | 585 |
| 570 | 484 | 554 | 585 | 507 | 537 | 578 | 492 | 523 | 585 | 585 | 554 | 554 | 585 | 523 | 578 | 570 | 507 | 617 | 578 | 554 | 554 | 585 | 523 | 570 | 554 | 523 | 554 | 523 | 554 | 539 | 554 |

**7 b). Detail of tank object from Alabama database (right half)**

FT.LOG;1 4-JAN-1984 11:21 Page 18

**** INVERSE TRANSFORMED DATA ****

INPUT FILES REAL : TEST0020 IMAGINARY : TEST0023

REAL PART LEFT HALF SCALE FACTOR = 9.9880

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 79 | 36 | 39 | -5 | -25 | 71 | 8 | 40 | 15 | -13 | 44 | -19 | 27 | 104 | 47 | 102 | 115 | 140 | 156 | 159 | 210 | 268 | 142 | 65 | 26 | 44 | 42 | 25 | -16 | -23 | -103 | -60 |
| 78 | 46 | 36 | 20 | 43 | 132 | 52 | 72 | 47 | 60 | 127 | 95 | 120 | 166 | 100 | 142 | 172 | 180 | 209 | 165 | 166 | 202 | 98 | 70 | 22 | 10 | 20 | -12 | -35 | -9 | -51 | -10 |
| 120 | 136 | 136 | 81 | 82 | 148 | 49 | 144 | 89 | 120 | 127 | 122 | 153 | 193 | 154 | 183 | 176 | 186 | 226 | 211 | 193 | 265 | 181 | 161 | 149 | 142 | 149 | 91 | 63 | 63 | -6 | 38 |
| 77 | 107 | 81 | 51 | 67 | 71 | -30 | -16 | -55 | -39 | 6 | 21 | 62 | 121 | 123 | 191 | 199 | 182 | 298 | 263 | 253 | 283 | 193 | 164 | 142 | 168 | 165 | 70 | 39 | 39 | 9 | 26 |
| 12 | 11 | -14 | 5 | 13 | 16 | -31 | -42 | -105 | -55 | -17 | -23 | 7 | 31 | 52 | 49 | 80 | 68 | 151 | 116 | 136 | 139 | 102 | 48 | 70 | 89 | 86 | 32 | 19 | 37 | 38 | 28 |
| -95 | -159 | -199 | -121 | -142 | -146 | -137 | -158 | -223 | -85 | -12 | 14 | 15 | 29 | 74 | 39 | 72 | 66 | 115 | 87 | 74 | 67 | 70 | 13 | 27 | 19 | 30 | -18 | -78 | -65 | -31 | -60 |
| -118 | -200 | -190 | -114 | -140 | -186 | -178 | -196 | -259 | -68 | 3 | -3 | 41 | 62 | 130 | 119 | 116 | 115 | 124 | 80 | 101 | 41 | 32 | 11 | 6 | 47 | 52 | 42 | -7 | -1 | 23 | -7 |
| -201 | -254 | -210 | -129 | -168 | -152 | -89 | -136 | -168 | 71 | 106 | 119 | 153 | 163 | 183 | 177 | 246 | 260 | 269 | 229 | 257 | 186 | 173 | 178 | 109 | 56 | 9 | 16 | -7 | -47 | 5 | -17 |
| -221 | -274 | -231 | -138 | -171 | -132 | -42 | -125 | -158 | 48 | 85 | 86 | 175 | 130 | 156 | 134 | 206 | 219 | 222 | 150 | 211 | 128 | 169 | 169 | 115 | 58 | 14 | 50 | 65 | 16 | 41 | 23 |
| -174 | -246 | -194 | -116 | -129 | -136 | -54 | -129 | -138 | 48 | 95 | 69 | 113 | 42 | 63 | 48 | 47 | 76 | 63 | -20 | 32 | -74 | -8 | 21 | -47 | -5 | -60 | 7 | 63 | 7 | 118 | 82 |
| -181 | -165 | -136 | -93 | -97 | -98 | -7 | -51 | -60 | 60 | 35 | 26 | 61 | -24 | 8 | -22 | -22 | 22 | -14 | -50 | 26 | -91 | -18 | -15 | -30 | 23 | -19 | 32 | 64 | -10 | 114 | 125 |
| -230 | -204 | -198 | -162 | -129 | -167 | -54 | -109 | -90 | 13 | 0 | 3 | 53 | -50 | -33 | -89 | -59 | -52 | -86 | -92 | 3 | -154 | -125 | -129 | -117 | -87 | -109 | -52 | -35 | -104 | 31 | 63 |
| -165 | -149 | -166 | -187 | -125 | -151 | -25 | -58 | -62 | 30 | -5 | 4 | 72 | -44 | -34 | -109 | -72 | -56 | -139 | -147 | -86 | -192 | -148 | -137 | -149 | -134 | -150 | -78 | -57 | -82 | 13 | 64 |
| -136 | -113 | -129 | -157 | -134 | -157 | -51 | -70 | -12 | 53 | 29 | 65 | 117 | 39 | 49 | 2 | 24 | 2 | -75 | -109 | -99 | -185 | -167 | -200 | -223 | -226 | -254 | -187 | -159 | -184 | -115 | -63 |
| -29 | 16 | 1 | -84 | -20 | -24 | 50 | 1 | 34 | 71 | -12 | 6 | 77 | 21 | 32 | 0 | 67 | 63 | -39 | -42 | -39 | -125 | -153 | -193 | -213 | -262 | -296 | -210 | -158 | -166 | -93 | -37 |
| -27 | -1 | -41 | -108 | -64 | -36 | 30 | -29 | -25 | -10 | -88 | -69 | -44 | -80 | -55 | -85 | -6 | 23 | -62 | -33 | -32 | -57 | -57 | -128 | -130 | -176 | -186 | -109 | -68 | -61 | -24 | 23 |
| 0 | -22 | -131 | -169 | -97 | -73 | -20 | -51 | -65 | -58 | -124 | -83 | -20 | -70 | -60 | -83 | -47 | -17 | -110 | -77 | -80 | -134 | -97 | -180 | -136 | -128 | -151 | -80 | -33 | -30 | -13 | -24 |
| 19 | -1 | -63 | -130 | -75 | -26 | 14 | -11 | -47 | -83 | -133 | -86 | -49 | -81 | -83 | -91 | -31 | 16 | -28 | 20 | -21 | -64 | -29 | -108 | -49 | -63 | -74 | -37 | 5 | 8 | 46 | 33 |
| 37 | 32 | -24 | -73 | -32 | 15 | 84 | 87 | 61 | 16 | -38 | 30 | 68 | 29 | 29 | -22 | 26 | 64 | 1 | 94 | 30 | -6 | 6 | -50 | 40 | -27 | -67 | -44 | -42 | 14 | -4 | 54 |
| 18 | -4 | -57 | -100 | -73 | -33 | 3 | -1 | 5 | -43 | -77 | -20 | 3 | 12 | 5 | -37 | -13 | -3 | -20 | 44 | -25 | -34 | -39 | -92 | -11 | -54 | -85 | -66 | -40 | 28 | 3 | 11 |
| 77 | 42 | -14 | -50 | -58 | -24 | -12 | -8 | -48 | -88 | -117 | -95 | -44 | -7 | -4 | 8 | 43 | 100 | 91 | 143 | 114 | 72 | 88 | 51 | 44 | -5 | -64 | -80 | -100 | -71 | -69 | -62 |
| 2 | 10 | -28 | -54 | -79 | -12 | -27 | 13 | 0 | -36 | -79 | -66 | -23 | 18 | -6 | -13 | 23 | 72 | 58 | 108 | 109 | 81 | 94 | 100 | 141 | 95 | 79 | 77 | 61 | 59 | 39 | 25 |
| -5 | 1 | -25 | -44 | -54 | -2 | -6 | 28 | 7 | -38 | -92 | -107 | -71 | -62 | -74 | -61 | -33 | -2 | 28 | 112 | 151 | 127 | 143 | 154 | 183 | 136 | 143 | 111 | 63 | 58 | 38 | 38 |
| -32 | -68 | -114 | -77 | -91 | -52 | -54 | -43 | -56 | -70 | -72 | -54 | -52 | -74 | -99 | -85 | -96 | -76 | -24 | 49 | 75 | 68 | 134 | 171 | 209 | 229 | 248 | 206 | 154 | 148 | 119 | 111 |
| -132 | -113 | -118 | -73 | -57 | -24 | -15 | 21 | 41 | 7 | -5 | 9 | -2 | -45 | -87 | -89 | -106 | -116 | -31 | 20 | 47 | 36 | 95 | 185 | 214 | 217 | 212 | 160 | 102 | 54 | 48 | 33 |
| -165 | -108 | -65 | -27 | -44 | -57 | -77 | -43 | -24 | -54 | -85 | -120 | -170 | -190 | -207 | -176 | -163 | -154 | -107 | -74 | -19 | -1 | 56 | 164 | 134 | 130 | 158 | 194 | 141 | 107 | 126 | 117 |
| -154 | -121 | -96 | -45 | -16 | -39 | -49 | -37 | 20 | 10 | -10 | -44 | -89 | -138 | -155 | -147 | -162 | -153 | -142 | -106 | -61 | -55 | -10 | 139 | 93 | 103 | 108 | 201 | 133 | 142 | 157 | 132 |
| -170 | -125 | -106 | -40 | -30 | -48 | -52 | -91 | -43 | -25 | -67 | -117 | -155 | -182 | -203 | -152 | -135 | -143 | -132 | -111 | -58 | -25 | -10 | 102 | 41 | 78 | 120 | 258 | 203 | 193 | 203 | 178 |
| -103 | -85 | -23 | 21 | 15 | 30 | 33 | -10 | 22 | 2 | -6 | -79 | -46 | -104 | -114 | -89 | -56 | -84 | -100 | -103 | -41 | -39 | 19 | 128 | 129 | 165 | 221 | 405 | 396 | 429 | 419 | 360 |
| -172 | -111 | -42 | 29 | 14 | 35 | 45 | 6 | 66 | 78 | 24 | -25 | -27 | -49 | -56 | -60 | -34 | -100 | -169 | -192 | -165 | -169 | -93 | 89 | 122 | 134 | 242 | 549 | 645 | 709 | 679 | 568 |
| -119 | -134 | -49 | 33 | -8 | 4 | 28 | -23 | 20 | 47 | 14 | -30 | -65 | -77 | -60 | -30 | -12 | -78 | -163 | -190 | -198 | -198 | -137 | 23 | 31 | 40 | 105 | 482 | 577 | 654 | 663 | 579 |
| -85 | -57 | -8 | 21 | -4 | -11 | 22 | -12 | -4 | 13 | -14 | -32 | -58 | -70 | -68 | -61 | -59 | -119 | -261 | -309 | -342 | -315 | -238 | -17 | 23 | 65 | 158 | 639 | 765 | 866 | 881 | 785 |
| -149 | -87 | -7 | 22 | 8 | 10 | 42 | 52 | 100 | 92 | 4 | -65 | -103 | -103 | -85 | -85 | -121 | -199 | -313 | -339 | -344 | -290 | -229 | -31 | 18 | 92 | 189 | 671 | 798 | 946 | 927 | 814 |
| -99 | -38 | 39 | 44 | 36 | 1 | 27 | 8 | 32 | 47 | -40 | -106 | -120 | -141 | -106 | -124 | -130 | -206 | -363 | -398 | -419 | -387 | -322 | -170 | -81 | 13 | 142 | 668 | 816 | 999 | 956 | 862 |
| -75 | -44 | 29 | -8 | -17 | -6 | 8 | 10 | 23 | 24 | -19 | -73 | -48 | -55 | -37 | -81 | -116 | -199 | -280 | -334 | -375 | -371 | -326 | -162 | -36 | 46 | 150 | 548 | 697 | 885 | 862 | 773 |
| -55 | -25 | 29 | -8 | -51 | -62 | -72 | -90 | -109 | -64 | -108 | -147 | -114 | -73 | -46 | -65 | -66 | -102 | -168 | -226 | -267 | -272 | -264 | -166 | -99 | -85 | -47 | 246 | 365 | 510 | 494 | 425 |
| -133 | -68 | 39 | 25 | 25 | -23 | -68 | -67 | -75 | -16 | -106 | -151 | -140 | -122 | -87 | -102 | -105 | -146 | -227 | -268 | -308 | -336 | -291 | -161 | -96 | -82 | -48 | 218 | 335 | 463 | 465 | 334 |
| -134 | -117 | -6 | 6 | 57 | -51 | -114 | -137 | -123 | -19 | -93 | -149 | -163 | -165 | -134 | -117 | -133 | -165 | -232 | -257 | -304 | -296 | -314 | -196 | -194 | -202 | -172 | 42 | 108 | 196 | 167 | 122 |
| -103 | -130 | -32 | -16 | 25 | -70 | -108 | -153 | -149 | -27 | -59 | -106 | -130 | -169 | -143 | -138 | -184 | -217 | -242 | -245 | -274 | -242 | -271 | -209 | -197 | -142 | -134 | 21 | 26 | 38 | 14 | 7 |
| -157 | -179 | -79 | -58 | -12 | -88 | -105 | -144 | -129 | -4 | -25 | -77 | -101 | -146 | -134 | -167 | -222 | -235 | -181 | -177 | -204 | -149 | -198 | -164 | -146 | -99 | -106 | -85 | -84 | -85 | -155 | -149 |
| -210 | -239 | -122 | -77 | -45 | -124 | -155 | -222 | -174 | -38 | -77 | -157 | -170 | -214 | -164 | -184 | -206 | -226 | -150 | -121 | -113 | -117 | -149 | -144 | -118 | -68 | -84 | -128 | -138 | -146 | -197 | -206 |
| -206 | -227 | -124 | -64 | -58 | -152 | -190 | -279 | -296 | -134 | -158 | -182 | -175 | -185 | -142 | -125 | -108 | -83 | -15 | -11 | -76 | -98 | -137 | -142 | -135 | -152 | -188 | -267 | -246 | -264 | -303 | -310 |
| -180 | -197 | -145 | -55 | -88 | -149 | -179 | -245 | -252 | -94 | -110 | -123 | -161 | -142 | -112 | -60 | -86 | -73 | -3 | -5 | -72 | -62 | -103 | -96 | -154 | -169 | -205 | -299 | -267 | -268 | -321 | -345 |
| -262 | -238 | -231 | -132 | -169 | -214 | -239 | -280 | -302 | -107 | -101 | -118 | -181 | -147 | -149 | -117 | -120 | -92 | -74 | -43 | -83 | -56 | -100 | -91 | -105 | -113 | -138 | -159 | -139 | -163 | -222 | -216 |
| -349 | -359 | -255 | -181 | -225 | -246 | -251 | -247 | -248 | -85 | -101 | -136 | -165 | -136 | -161 | -147 | -167 | -169 | -114 | -94 | -126 | -103 | -110 | -109 | -83 | -54 | -12 | -38 | 14 | 11 | -52 | -52 |
| -291 | -321 | -229 | -181 | -210 | -228 | -247 | -229 | -257 | -118 | -82 | -103 | -140 | -86 | -94 | -63 | -106 | -82 | -24 | -8 | -64 | 8 | -49 | -79 | -76 | -52 | -18 | -33 | 49 | 52 | -33 | 10 |
| -336 | -336 | -236 | -185 | -239 | -262 | -266 | -199 | -236 | -122 | -77 | -62 | -127 | -37 | -19 | 16 | -8 | -16 | 62 | 111 | 83 | 110 | 73 | 16 | 9 | 36 | 92 | 52 | 110 | 74 | 53 | 58 |
| -329 | -313 | -234 | -216 | -239 | -246 | -264 | -197 | -233 | -127 | -112 | -76 | -145 | -48 | -19 | 20 | 18 | 15 | 74 | 144 | 79 | 125 | 78 | -3 | -10 | -13 | 61 | 6 | 41 | 75 | 61 | 72 |
| -291 | -197 | -164 | -147 | -103 | -143 | -193 | -162 | -193 | -82 | -104 | -52 | -113 | -59 | -7 | 40 | 69 | 44 | 21 | 92 | 76 | 74 | 75 | -4 | -4 | -41 | 32 | -11 | 57 | 84 | 108 | 112 |
| -206 | -143 | -164 | -106 | -45 | -113 | -115 | -66 | -94 | 19 | 18 | 110 | 36 | 94 | 102 | 108 | 117 | 96 | 100 | 126 | 110 | 127 | 80 | -9 | -11 | -43 | 6 | -27 | 49 | 62 | 72 | 84 |
| -174 | -64 | -94 | -145 | -42 | -110 | -160 | -50 | -30 | 52 | 18 | 93 | 12 | 31 | 29 | 2 | 52 | -4 | -36 | 16 | 32 | 53 | 4 | -76 | -32 | -105 | -36 | -48 | 0 | -21 | 20 | 83 |
| -227 | -134 | -129 | -178 | -105 | -119 | -178 | -67 | -73 | 16 | -27 | 30 | -10 | 59 | 72 | 26 | 110 | 89 | 95 | 128 | 119 | 129 | 45 | -48 | -3 | -108 | -31 | -65 | -3 | -25 | 31 | 120 |
| -148 | -5 | -32 | -21 | 32 | 22 | -41 | 15 | 16 | 65 | -55 | 65 | 53 | 87 | 119 | 96 | 130 | 89 | 96 | 145 | 118 | 85 | 32 | -32 | -5 | -113 | -28 | -84 | -8 | -19 | 27 | 101 |
| -14 | 87 | 20 | 31 | 80 | 75 | 25 | 47 | 9 | 55 | 6 | 69 | 20 | 53 | 80 | 63 | 134 | 126 | 122 | 174 | 197 | 176 | 114 | 72 | 77 | -29 | 23 | -7 | 28 | -7 | 22 | 63 |
| 13 | 103 | -29 | -53 | -1 | 13 | -35 | -13 | -86 | -39 | -69 | 30 | 20 | 16 | 12 | -17 | 36 | 24 | 84 | 77 | 94 | 65 | 72 | 22 | 95 | 12 | 124 | 6 | 45 | 19 | 56 | 74 |

**8 a). Inverse transform of modulus plane from 8 by 8 square, phase plane from tank object (appendix 7). (left half)**

FT.LOG;1 4-JAN-1984 11:21 Page 19

**** INVERSE TRANSFORMED DATA ****

INPUT FILES REAL : TEST0020 IMAGINARY : TEST0023

REAL PART RIGHT HALF SCALE FACTOR = 9.9880

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| -6 | 110 | 55 | 22 | 33 | -3 | 3 | 37 | 8 | 6 | -97 | -62 | -70 | -104 | -51 | -38 | -35 | -5 | -18 | -23 | 23 | 48 | 76 | 100 | 74 | 103 | 94 | 125 | 146 | 136 | 48 | 73 |
| 16 | 108 | 88 | 58 | 50 | 37 | 9 | 53 | 15 | 9 | -46 | 7 | 9 | -12 | 24 | 62 | 25 | 56 | 43 | 47 | 36 | 55 | 58 | 100 | 80 | 131 | 109 | 122 | 114 | 162 | 70 | 121 |
| 51 | 107 | 83 | 73 | 87 | 29 | 5 | 116 | 54 | 14 | -46 | -8 | -45 | -74 | 38 | 81 | -13 | -10 | 17 | -1 | -13 | 4 | 15 | 24 | -2 | 48 | 73 | 88 | 110 | 216 | 110 | 179 |
| 45 | 77 | 42 | 73 | 121 | 87 | 65 | 150 | 132 | 122 | 79 | 88 | 39 | -24 | 62 | 58 | -22 | -33 | -23 | 1 | -10 | 2 | 39 | 65 | 50 | 99 | 106 | 89 | 90 | 179 | 128 | 184 |
| 61 | 96 | 75 | 104 | 127 | 81 | 68 | 113 | 135 | 144 | 117 | 122 | 127 | 87 | 162 | 150 | 115 | 53 | 66 | 79 | 95 | 38 | 55 | 119 | 116 | 129 | 138 | 115 | 101 | 123 | 118 | 184 |
| -31 | -37 | -41 | -15 | 8 | 53 | 65 | 75 | 126 | 140 | 147 | 178 | 177 | 144 | 193 | 170 | 130 | 40 | 73 | 81 | 95 | 31 | 44 | 120 | 149 | 182 | 169 | 143 | 126 | 108 | 143 | 190 |
| -1 | 25 | 40 | 92 | 127 | 147 | 155 | 192 | 222 | 205 | 220 | 195 | 171 | 133 | 209 | 177 | 129 | 96 | 132 | 94 | 116 | 74 | 34 | 143 | 204 | 196 | 151 | 137 | 156 | 150 | 189 | 247 |
| -13 | 29 | 59 | 101 | 96 | 112 | 116 | 149 | 159 | 133 | 134 | 143 | 113 | 89 | 179 | 152 | 138 | 129 | 165 | 129 | 124 | 116 | 125 | 201 | 273 | 200 | 156 | 151 | 152 | 154 | 202 | 205 |
| -17 | 5 | 17 | 52 | 26 | 27 | 21 | 62 | 82 | 81 | 76 | 115 | 92 | 53 | 132 | 122 | 93 | 62 | 127 | 123 | 87 | 46 | 80 | 142 | 208 | 138 | 84 | 45 | 21 | 12 | 128 | 126 |
| 50 | 64 | 54 | 88 | 31 | 13 | 70 | 35 | 95 | 99 | 141 | 159 | 149 | 149 | 216 | 170 | 168 | 111 | 151 | 127 | 60 | 19 | 36 | 63 | 150 | 83 | 59 | 37 | 28 | 28 | 169 | 125 |
| 87 | 97 | 63 | 75 | 20 | -23 | 36 | -8 | -2 | 20 | 3 | 18 | -21 | 8 | 96 | 35 | 26 | -9 | 24 | 21 | -6 | -25 | 4 | 23 | 76 | 42 | 50 | 11 | 15 | -26 | 83 | 36 |
| 63 | 87 | 95 | 69 | 40 | 34 | 94 | 6 | 34 | 101 | 78 | 62 | 87 | 61 | 66 | 24 | 74 | 8 | 1 | -7 | -4 | -22 | 14 | 4 | 11 | -45 | -17 | -41 | -36 | -132 | -66 | -108 |
| 53 | 117 | 113 | 77 | 94 | 53 | 91 | 25 | 24 | 28 | -26 | -36 | -37 | -73 | -41 | -56 | -43 | -113 | -44 | -59 | -39 | -51 | -31 | -56 | -44 | -60 | -3 | -73 | -44 | -127 | -66 | -79 |
| -50 | 11 | 32 | 16 | 10 | -36 | 22 | -4 | -23 | -5 | -66 | -85 | -114 | -138 | -84 | -110 | -142 | -181 | -138 | -140 | -167 | -106 | -77 | -137 | -134 | -120 | -66 | -100 | -64 | -102 | -73 | -125 |
| -25 | -2 | 65 | 39 | 8 | -63 | -57 | -103 | -106 | -119 | -104 | -153 | -163 | -174 | -118 | -123 | -138 | -183 | -154 | -187 | -235 | -128 | -107 | -203 | -187 | -163 | -58 | -101 | -61 | -69 | -84 | -112 |
| 57 | 66 | 119 | 47 | 0 | -58 | -71 | -123 | -152 | -176 | -173 | -187 | -178 | -169 | -122 | -121 | -104 | -149 | -114 | -122 | -186 | -78 | -46 | -183 | -170 | -146 | -42 | -65 | -77 | -98 | -134 | -169 |
| 28 | -26 | 4 | -46 | -81 | -120 | -128 | -172 | -140 | -153 | -87 | -137 | -137 | -113 | -99 | -82 | -83 | -151 | -142 | -128 | -186 | -41 | -13 | -131 | -98 | -10 | 48 | 20 | -16 | -55 | -103 | -141 |
| 67 | 10 | 63 | 7 | -13 | -75 | -106 | -153 | -151 | -156 | -127 | -178 | -191 | -182 | -164 | -136 | -119 | -157 | -129 | -118 | -172 | -22 | 3 | -119 | -76 | -27 | 48 | 5 | -17 | -54 | -131 | -160 |
| 58 | -31 | 39 | 32 | 36 | -33 | -123 | -97 | -157 | -145 | -130 | -171 | -206 | -225 | -212 | -156 | -189 | -178 | -129 | -74 | -54 | 47 | 81 | -34 | -23 | 26 | 60 | 0 | -31 | -128 | -176 | -187 |
| 45 | -35 | 36 | 23 | 25 | -65 | -129 | -87 | -99 | -145 | -94 | -148 | -116 | -142 | -116 | -81 | -105 | -124 | -48 | -35 | 9 | 63 | 102 | -18 | 11 | 40 | 78 | 21 | -2 | -88 | -128 | -159 |
| -58 | -38 | -44 | -16 | -41 | -121 | -137 | -149 | -178 | -234 | -174 | -196 | -216 | -185 | -138 | -121 | -180 | -154 | -76 | -77 | -78 | -14 | 14 | -37 | 40 | 130 | 123 | 56 | 58 | 11 | -22 | -40 |
| 11 | -41 | -27 | -48 | -139 | -203 | -247 | -231 | -259 | -307 | -226 | -226 | -187 | -132 | -94 | -86 | -101 | -82 | -38 | -41 | -55 | 10 | 56 | 15 | 51 | 110 | 128 | 104 | 55 | 26 | -63 | -130 |
| 41 | -15 | -7 | -45 | -117 | -159 | -209 | -166 | -185 | -269 | -190 | -187 | -129 | -92 | -37 | -34 | -47 | -31 | 47 | 45 | 31 | 63 | 85 | 68 | 104 | 136 | 139 | 103 | 45 | 15 | -51 | -65 |
| 86 | 86 | 71 | 56 | -11 | -61 | -70 | -20 | -51 | -99 | -75 | -79 | -16 | 22 | 57 | 42 | 10 | 44 | 111 | 118 | 127 | 97 | 100 | 129 | 167 | 196 | 151 | 122 | 58 | -1 | -13 | -17 |
| -1 | -8 | -9 | 22 | -43 | -81 | -81 | -40 | -37 | -82 | -51 | -84 | -61 | -30 | -29 | -30 | -57 | -31 | 31 | 31 | 61 | 44 | 58 | 155 | 181 | 192 | 136 | 68 | 16 | -30 | -43 | -54 |
| 88 | 111 | 91 | 92 | 9 | -26 | -37 | -11 | -53 | -103 | -109 | -108 | -117 | -102 | -79 | -79 | -96 | -49 | 5 | 29 | 57 | 33 | 16 | 123 | 135 | 109 | 71 | 30 | -10 | -27 | -55 | -38 |
| 64 | 69 | 30 | 40 | -68 | -101 | -128 | -152 | -175 | -210 | -164 | -172 | -174 | -127 | -87 | -92 | -98 | -43 | -11 | 0 | 44 | 21 | -29 | 70 | 89 | 107 | 58 | 34 | -15 | -27 | -17 | 4 |
| 146 | 134 | 43 | 5 | -116 | -102 | -102 | -153 | -147 | -184 | -144 | -137 | -124 | -74 | -45 | -41 | -40 | -28 | -32 | -33 | -9 | -4 | -53 | 49 | 48 | 82 | 47 | 57 | 23 | 39 | 50 | 62 |
| 267 | 138 | 72 | 61 | -62 | -33 | -75 | -130 | -122 | -109 | -77 | -78 | -118 | -88 | -105 | -92 | -100 | -85 | -97 | -51 | -58 | -44 | -73 | 20 | 3 | 10 | -26 | 14 | -45 | 5 | 55 | 69 |
| 362 | 226 | 147 | 108 | -40 | -43 | -123 | -146 | -126 | -82 | -26 | -50 | -59 | -51 | -104 | -72 | -66 | -56 | -68 | -4 | 26 | -7 | -42 | 45 | 1 | -17 | -47 | -5 | -86 | -60 | 32 | 11 |
| 364 | 200 | 147 | 142 | 4 | 26 | -24 | -43 | -38 | 17 | 125 | 102 | 82 | 52 | 3 | 1 | -19 | 15 | -5 | 36 | 66 | 40 | 10 | 77 | 98 | 81 | 13 | 85 | 16 | -2 | 112 | 109 |
| 521 | 295 | 222 | 245 | 59 | 50 | -33 | -73 | -104 | -28 | 90 | 53 | -22 | -55 | -136 | -112 | -115 | -81 | -116 | -77 | -19 | -38 | -35 | 56 | 51 | 41 | 13 | 91 | 51 | 30 | 130 | 118 |
| 586 | 321 | 212 | 207 | 21 | -10 | -136 | -149 | -163 | -106 | 9 | 1 | -73 | -111 | -169 | -153 | -126 | -67 | -86 | -55 | 0 | 28 | 48 | 126 | 142 | 92 | 34 | 82 | 49 | 47 | 86 | 42 |
| 651 | 346 | 211 | 184 | -3 | -38 | -153 | -151 | -184 | -103 | 43 | 52 | 21 | -21 | -50 | -51 | -19 | 61 | 22 | 70 | 88 | 73 | 106 | 174 | 171 | 117 | 64 | 133 | 92 | 120 | 170 | 115 |
| 527 | 244 | 157 | 119 | 14 | -25 | -107 | -100 | -124 | -35 | 86 | 31 | 70 | -3 | -5 | -36 | -46 | 39 | 10 | 12 | 64 | 11 | 85 | 140 | 133 | 111 | 75 | 132 | 99 | 153 | 209 | 137 |
| 290 | 86 | 69 | 82 | 58 | 72 | 37 | 90 | 61 | 122 | 233 | 198 | 245 | 130 | 100 | 67 | 21 | 93 | 33 | 36 | 73 | 19 | 114 | 208 | 189 | 182 | 148 | 234 | 192 | 247 | 276 | 212 |
| 166 | 36 | 20 | 60 | 18 | -1 | -20 | 43 | 27 | 94 | 179 | 146 | 184 | 106 | 117 | 50 | -7 | 60 | 27 | 23 | 57 | -24 | 67 | 183 | 190 | 146 | 122 | 173 | 146 | 184 | 223 | 157 |
| 43 | -7 | 29 | 24 | 25 | 23 | 28 | 95 | 82 | 119 | 197 | 151 | 204 | 116 | 149 | 75 | 43 | 97 | 66 | 35 | 103 | 32 | 82 | 202 | 267 | 222 | 183 | 184 | 186 | 193 | 258 | 215 |
| -5 | -40 | -31 | -34 | -22 | -6 | 66 | 98 | 96 | 116 | 204 | 161 | 198 | 113 | 142 | 62 | 32 | 86 | 63 | 19 | 74 | 40 | 97 | 225 | 324 | 280 | 235 | 252 | 246 | 240 | 303 | 303 |
| -115 | -133 | -117 | -95 | -62 | -45 | -8 | 40 | 34 | 42 | 96 | 73 | 93 | 45 | 89 | 69 | 43 | 106 | 79 | 48 | 92 | 37 | 94 | 247 | 316 | 275 | 236 | 254 | 250 | 250 | 321 | 308 |
| -154 | -155 | -172 | -192 | -127 | -67 | 21 | 67 | 113 | 94 | 140 | 123 | 170 | 136 | 167 | 101 | 69 | 130 | 100 | 108 | 159 | 109 | 149 | 300 | 349 | 313 | 246 | 280 | 220 | 208 | 293 | 273 |
| -269 | -274 | -213 | -226 | -153 | -111 | 16 | 62 | 82 | 94 | 157 | 106 | 154 | 161 | 193 | 141 | 113 | 158 | 101 | 109 | 152 | 86 | 91 | 298 | 308 | 256 | 199 | 261 | 217 | 196 | 316 | 333 |
| -285 | -226 | -164 | -195 | -97 | -72 | 13 | 102 | 136 | 108 | 120 | 82 | 135 | 123 | 146 | 112 | 37 | 74 | 43 | 46 | 125 | 80 | 36 | 255 | 270 | 264 | 221 | 266 | 240 | 180 | 281 | 357 |
| -141 | -110 | -86 | -98 | -49 | -34 | 21 | 118 | 148 | 121 | 171 | 199 | 199 | 197 | 168 | 127 | 71 | 85 | 54 | 37 | 82 | 76 | -2 | 263 | 307 | 279 | 240 | 267 | 233 | 160 | 212 | 310 |
| 29 | 46 | 42 | -14 | -1 | -35 | -36 | 55 | 65 | 22 | 43 | 46 | 37 | 87 | 107 | 94 | 62 | 78 | 64 | 30 | 58 | 44 | -33 | 231 | 278 | 193 | 173 | 181 | 175 | 135 | 164 | 229 |
| 113 | 148 | 164 | 96 | 114 | 67 | 52 | 151 | 152 | 79 | 51 | 49 | 53 | 91 | 113 | 104 | 69 | 77 | 39 | 36 | 36 | 25 | -68 | 150 | 157 | 92 | 96 | 101 | 116 | 131 | 145 | 229 |
| 148 | 185 | 193 | 144 | 149 | 113 | 105 | 159 | 151 | 56 | 28 | 41 | 20 | 82 | 90 | 117 | 111 | 158 | 113 | 106 | 75 | 94 | 26 | 258 | 172 | 68 | 60 | 112 | 143 | 172 | 83 | 162 |
| 167 | 208 | 233 | 170 | 200 | 175 | 129 | 219 | 224 | 117 | 66 | 74 | 74 | 119 | 147 | 157 | 120 | 152 | 145 | 158 | 126 | 126 | 53 | 286 | 211 | 108 | 107 | 97 | 131 | 167 | 69 | 128 |
| 179 | 207 | 233 | 185 | 222 | 191 | 133 | 191 | 188 | 132 | 99 | 97 | 90 | 162 | 150 | 129 | 153 | 151 | 162 | 199 | 160 | 169 | 50 | 264 | 167 | 58 | 70 | -21 | 2 | 65 | -17 | 77 |
| 144 | 161 | 201 | 166 | 173 | 121 | 45 | 69 | 122 | 125 | 142 | 117 | 128 | 199 | 212 | 212 | 274 | 236 | 218 | 192 | 174 | 174 | 75 | 275 | 208 | 111 | 83 | -12 | 108 | 143 | 68 | 146 |
| 132 | 133 | 213 | 162 | 215 | 176 | 125 | 125 | 193 | 127 | 134 | 132 | 165 | 196 | 220 | 223 | 223 | 158 | 160 | 143 | 124 | 115 | 49 | 159 | 75 | 70 | 138 | 10 | 105 | 228 | 43 | 97 |
| 219 | 195 | 273 | 233 | 237 | 204 | 194 | 139 | 218 | 157 | 200 | 160 | 143 | 209 | 213 | 200 | 216 | 116 | 94 | 53 | -2 | 18 | -22 | 56 | -6 | -12 | 11 | -59 | -2 | 157 | -41 | -16 |
| 195 | 200 | 321 | 274 | 277 | 228 | 196 | 167 | 236 | 121 | 143 | 73 | 113 | 180 | 200 | 223 | 257 | 162 | 138 | 124 | 84 | 67 | 50 | 81 | -19 | -56 | -78 | -86 | -52 | 73 | -28 | -29 |
| 95 | 125 | 218 | 164 | 169 | 130 | 130 | 157 | 265 | 212 | 190 | 139 | 193 | 204 | 218 | 237 | 268 | 170 | 178 | 206 | 165 | 176 | 197 | 178 | 77 | 71 | 51 | 0 | 31 | 127 | 35 | 43 |
| 117 | 75 | 163 | 114 | 140 | 114 | 106 | 65 | 136 | 128 | 160 | 112 | 92 | 122 | 61 | 80 | 148 | 73 | 119 | 117 | 72 | 106 | 122 | 110 | 78 | 100 | 92 | 2 | 11 | 86 | -2 | 33 |

**8 b). Inverse transform of modulus plane from 8 by 8 square, phase plane from tank object (appendix 7) (right half)**

'alsh transform of pictures

Picture name : SQFL0101 Scale factor = 1.0000 Extreme Left

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 999 | 999 | 999 | 999 | 783 | 783 | 783 | 783 | -98 | -98 | -98 | -98 | 108 | 108 | 108 | 108 |
| 999 | 999 | 999 | 999 | 783 | 783 | 783 | 783 | -98 | -98 | -98 | -98 | 108 | 108 | 108 | 108 |
| 999 | 999 | 999 | 999 | 783 | 783 | 783 | 783 | -98 | -98 | -98 | -98 | 108 | 108 | 108 | 108 |
| 999 | 999 | 999 | 999 | 783 | 783 | 783 | 783 | -98 | -98 | -98 | -98 | 108 | 108 | 108 | 108 |
| 777 | 777 | 777 | 777 | 609 | 609 | 609 | 609 | -76 | -76 | -76 | -76 | 84 | 84 | 84 | 84 |
| 777 | 777 | 777 | 777 | 609 | 609 | 609 | 609 | -76 | -76 | -76 | -76 | 84 | 84 | 84 | 84 |
| 777 | 777 | 777 | 777 | 609 | 609 | 609 | 609 | -76 | -76 | -76 | -76 | 84 | 84 | 84 | 84 |
| 777 | 777 | 777 | 777 | 609 | 609 | 609 | 609 | -76 | -76 | -76 | -76 | 84 | 84 | 84 | 84 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | -11 | -11 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | -11 | -11 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | -11 | -11 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | -11 | -11 |
| 111 | 111 | 111 | 111 | 87 | 87 | 87 | 87 | -10 | -10 | -10 | -10 | 12 | 12 | 12 | 12 |
| 111 | 111 | 111 | 111 | 87 | 87 | 87 | 87 | -10 | -10 | -10 | -10 | 12 | 12 | 12 | 12 |
| 111 | 111 | 111 | 111 | 87 | 87 | 87 | 87 | -10 | -10 | -10 | -10 | 12 | 12 | 12 | 12 |
| 111 | 111 | 111 | 111 | 87 | 87 | 87 | 87 | -10 | -10 | -10 | -10 | 12 | 12 | 12 | 12 |
| 111 | 111 | 111 | 111 | 87 | 87 | 87 | 87 | -10 | -10 | -10 | -10 | 12 | 12 | 12 | 12 |
| 111 | 111 | 111 | 111 | 87 | 87 | 87 | 87 | -10 | -10 | -10 | -10 | 12 | 12 | 12 | 12 |
| 111 | 111 | 111 | 111 | 87 | 87 | 87 | 87 | -10 | -10 | -10 | -10 | 12 | 12 | 12 | 12 |
| 111 | 111 | 111 | 111 | 87 | 87 | 87 | 87 | -10 | -10 | -10 | -10 | 12 | 12 | 12 | 12 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | -11 | -11 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | -11 | -11 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | -11 | -11 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | -11 | -11 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | -11 | -11 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | -11 | -11 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | -11 | -11 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | -11 | -11 |
| 111 | 111 | 111 | 111 | 87 | 87 | 87 | 87 | -10 | -10 | -10 | -10 | 12 | 12 | 12 | 12 |
| 111 | 111 | 111 | 111 | 87 | 87 | 87 | 87 | -10 | -10 | -10 | -10 | 12 | 12 | 12 | 12 |
| 111 | 111 | 111 | 111 | 87 | 87 | 87 | 87 | -10 | -10 | -10 | -10 | 12 | 12 | 12 | 12 |
| 111 | 111 | 111 | 111 | 87 | 87 | 87 | 87 | -10 | -10 | -10 | -10 | 12 | 12 | 12 | 12 |
| 111 | 111 | 111 | 111 | 87 | 87 | 87 | 87 | -10 | -10 | -10 | -10 | 12 | 12 | 12 | 12 |
| 111 | 111 | 111 | 111 | 87 | 87 | 87 | 87 | -10 | -10 | -10 | -10 | 12 | 12 | 12 | 12 |
| 111 | 111 | 111 | 111 | 87 | 87 | 87 | 87 | -10 | -10 | -10 | -10 | 12 | 12 | 12 | 12 |
| 111 | 111 | 111 | 111 | 87 | 87 | 87 | 87 | -10 | -10 | -10 | -10 | 12 | 12 | 12 | 12 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | -11 | -11 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | -11 | -11 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | -11 | -11 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | -11 | -11 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | -11 | -11 |
| -110 | -110 | -110 | -110 | -86 | -86 | -86 | -86 | 11 | 11 | 11 | 11 | -11 | -11 | [illegible] | [illegible] |

9 a). Walsh transform of square at 01, 01

## Walsh transform of pictures

Picture name : SQFL1020 Scale factor = 1.0000 Extreme Left

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 999 | 999 | 558 | 558 | -989 | -989 | -548 | -548 | 333 | 333 | -98 | -98 | -323 | -323 | 108 | 108 |
| 999 | 999 | 558 | 558 | -989 | -989 | -548 | -548 | 333 | 333 | -98 | -98 | -323 | -323 | 108 | 108 |
| -998 | -998 | -557 | -557 | 990 | 990 | 549 | 549 | -332 | -332 | 99 | 99 | 324 | 324 | -107 | -107 |
| -998 | -998 | -557 | -557 | 990 | 990 | 549 | 549 | -332 | -332 | 99 | 99 | 324 | 324 | -107 | -107 |
| -110 | -110 | -61 | -61 | 110 | 110 | 61 | 61 | -36 | -36 | 11 | 11 | 36 | 36 | -11 | -11 |
| -110 | -110 | -61 | -61 | 110 | 110 | 61 | 61 | -36 | -36 | 11 | 11 | 36 | 36 | -11 | -11 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |
| -776 | -776 | -433 | -433 | 770 | 770 | 427 | 427 | -258 | -258 | 77 | 77 | 252 | 252 | -83 | -83 |
| -776 | -776 | -433 | -433 | 770 | 770 | 427 | 427 | -258 | -258 | 77 | 77 | 252 | 252 | -83 | -83 |
| 777 | 777 | 434 | 434 | -769 | -769 | -426 | -426 | 259 | 259 | -76 | -76 | -251 | -251 | 84 | 84 |
| 777 | 777 | 434 | 434 | -769 | -769 | -426 | -426 | 259 | 259 | -76 | -76 | -251 | -251 | 84 | 84 |
| -110 | -110 | -61 | -61 | 110 | 110 | 61 | 61 | -36 | -36 | 11 | 11 | 36 | 36 | -11 | -11 |
| -110 | -110 | -61 | -61 | 110 | 110 | 61 | 61 | -36 | -36 | 11 | 11 | 36 | 36 | -11 | -11 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |
| -110 | -110 | -61 | -61 | 110 | 110 | 61 | 61 | -36 | -36 | 11 | 11 | 36 | 36 | -11 | -11 |
| -110 | -110 | -61 | -61 | 110 | 110 | 61 | 61 | -36 | -36 | 11 | 11 | 36 | 36 | -11 | -11 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |
| -110 | -110 | -61 | -61 | 110 | 110 | 61 | 61 | -36 | -36 | 11 | 11 | 36 | 36 | -11 | -11 |
| -110 | -110 | -61 | -61 | 110 | 110 | 61 | 61 | -36 | -36 | 11 | 11 | 36 | 36 | -11 | -11 |
| -110 | -110 | -61 | -61 | 110 | 110 | 61 | 61 | -36 | -36 | 11 | 11 | 36 | 36 | -11 | -11 |
| -110 | -110 | -61 | -61 | 110 | 110 | 61 | 61 | -36 | -36 | 11 | 11 | 36 | 36 | -11 | -11 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |
| -110 | -110 | -61 | -61 | 110 | 110 | 61 | 61 | -36 | -36 | 11 | 11 | 36 | 36 | -11 | -11 |
| -110 | -110 | -61 | -61 | 110 | 110 | 61 | 61 | -36 | -36 | 11 | 11 | 36 | 36 | -11 | -11 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |
| -110 | -110 | -61 | -61 | 110 | 110 | 61 | 61 | -36 | -36 | 11 | 11 | 36 | 36 | -11 | -11 |
| -110 | -110 | -61 | -61 | 110 | 110 | 61 | 61 | -36 | -36 | 11 | 11 | 36 | 36 | -11 | -11 |
| -110 | -110 | -61 | -61 | 110 | 110 | 61 | 61 | -36 | -36 | 11 | 11 | 36 | 36 | -11 | -11 |
| -110 | -110 | -61 | -61 | 110 | 110 | 61 | 61 | -36 | -36 | 11 | 11 | 36 | 36 | -11 | -11 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |
| 111 | 111 | 62 | 62 | -109 | -109 | -60 | -60 | 37 | 37 | -10 | -10 | -35 | -35 | 12 | 12 |

9 b). Walsh transform of square at 10, 20

C.LOG11 29-FEB-1994 11:21 Page 2

| F/S | Fourier Modulus | Walsh n= 1 | 26 | 51 | 76 | 101 | 126 | 151 | 176 | 201 | 226 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 5376.0000 | 5376 | 5376 | 5376 | 5376 | 5376 | 5376 | 5376 | 5376 | 5376 | 5376 |
| 2 | 5316.8242 | 5376 | 5376 | 5376 | 5376 | 5376 | -3840 | -5376 | -5376 | -5376 | -5376 |
| 3 | 5141.6313 | 5376 | 5376 | 1792 | -5376 | -5376 | -5376 | -5376 | -3328 | 5376 | 5376 |
| 4 | 4857.3164 | 5376 | 5376 | 1792 | -5376 | -5376 | 3840 | 5376 | 3328 | -5376 | -5376 |
| 5 | 4475.0137 | 5376 | -1792 | -5376 | -5376 | 5376 | 5376 | -256 | -5376 | -5376 | 5376 |
| 6 | 4009.5767 | 5376 | -1792 | -5376 | -5376 | 5376 | -3840 | 256 | 5376 | 5376 | -5376 |
| 7 | 3473.8377 | 5376 | -1792 | -1792 | 5376 | -5376 | -5376 | 256 | 3328 | -5376 | 5376 |
| 8 | 2903.0344 | 5376 | -1792 | -1792 | 5376 | -5376 | 3840 | -256 | -3328 | 5376 | -5376 |
| 9 | 2303.3909 | 2816 | -5376 | 5376 | -2816 | -768 | 4352 | -5376 | 4864 | -1280 | -2304 |
| 10 | 1701.6482 | 2816 | -5376 | 5376 | -2816 | -768 | -2816 | 5376 | -4864 | 1280 | 2304 |
| 11 | 1118.8510 | 2816 | -5376 | 1792 | 2816 | 768 | -4352 | 5376 | -2816 | -1280 | -2304 |
| 12 | 574.4760 | 2816 | -5376 | 1792 | 2816 | 768 | 2816 | -5376 | 2816 | 1280 | 2304 |
| 13 | 85.6071 | 2816 | 1792 | -5376 | 2816 | -768 | 4352 | 256 | -4864 | 1280 | -2304 |
| 14 | 333.7611 | 2816 | 1792 | -5376 | 2816 | -768 | -2816 | -256 | 4864 | -1280 | 2304 |
| 15 | 673.2534 | 2816 | 1792 | -1792 | -2816 | 768 | -4352 | -256 | 2816 | 1280 | -2304 |
| 16 | 926.4570 | 2816 | 1792 | -1792 | -2816 | 768 | 2816 | 256 | -2816 | -1280 | 2304 |
| 17 | 1091.0649 | -1280 | 2304 | 2304 | -1280 | -2816 | 256 | 2816 | 768 | -2816 | -1792 |
| 18 | 1168.8177 | -1280 | 2304 | 2304 | -1280 | -2816 | 1280 | -2816 | -768 | 2816 | 1792 |
| 19 | 1165.2439 | -1280 | 2304 | -1280 | 1280 | 2816 | -256 | -2816 | 1280 | -2816 | -1792 |
| 20 | 1089.2279 | -1280 | 2304 | -1280 | 1280 | 2816 | -1280 | 2816 | -1280 | 2816 | 1792 |
| 21 | 952.4285 | -1280 | 1280 | -2304 | 1280 | -2816 | 256 | 256 | -768 | 2816 | -1792 |
| 22 | 768.5728 | -1280 | 1280 | -2304 | 1280 | -2816 | 1280 | -256 | 768 | -2816 | 1792 |
| 23 | 552.6802 | -1280 | 1280 | 1280 | -1280 | 2816 | -256 | -256 | -1280 | 2816 | -1792 |
| 24 | 320.2470 | -1280 | 1280 | 1280 | -1280 | 2816 | -1280 | 256 | 1280 | -2816 | 1792 |
| 25 | 86.4406 | 1280 | -2304 | 2304 | -1280 | -768 | 1280 | -2816 | 1280 | -1280 | -1280 |
| 26 | 134.6566 | 1280 | -2304 | 2304 | -1280 | -768 | 256 | 2816 | -1280 | 1280 | 1280 |
| 27 | 330.72[illegible] | [illegible] | 2304 | [illegible] | 1280 | 768 | -1280 | 2816 | 768 | -1280 | -1280 |
| 28 | 491.72[illegible] | 1280 | -2304 | 2304 | 1280 | 768 | -256 | -2816 | -768 | 1280 | 1280 |
| 29 | 610.3512 | 1280 | -1280 | -2304 | 1280 | -768 | 1280 | -256 | -1280 | 1280 | -1280 |
| 30 | 682.2183 | 1280 | -1280 | -2304 | 1280 | -768 | 256 | 256 | 1280 | -1280 | 1280 |
| 31 | 705.9691 | 1280 | 1280 | 1280 | -1280 | 768 | -1280 | 256 | -768 | 1280 | -1280 |
| 32 | 683.1656 | 1280 | -1280 | 1280 | -1280 | 768 | -256 | -256 | 768 | -1280 | 1280 |
| 33 | 618.0385 | 768 | -256 | -256 | 768 | -1280 | 1280 | -1280 | 1280 | -768 | 256 |
| 34 | 517.1023 | 768 | -256 | -256 | 768 | -1280 | 256 | 1280 | -1280 | 768 | -256 |
| 35 | 388.6554 | 768 | -256 | -768 | -768 | 1280 | -1280 | 1280 | 768 | -768 | 256 |
| 36 | 242.2064 | 768 | -256 | -768 | -768 | 1280 | -256 | -1280 | -768 | 768 | -256 |
| 37 | 87.8553 | 768 | 768 | 256 | -768 | -1280 | 1280 | 256 | -1280 | 768 | 256 |
| 38 | 64.3316 | 768 | 768 | 256 | -768 | -1280 | 256 | -256 | 1280 | -768 | -256 |
| 39 | 204.9173 | 768 | 768 | 768 | 768 | 1280 | -1280 | -256 | -768 | 768 | 256 |
| 40 | 325.5279 | 768 | 768 | 768 | 768 | 1280 | -256 | 256 | 768 | -768 | -256 |
| 41 | 419.7963 | -768 | 256 | -256 | 768 | 768 | 256 | 1280 | 768 | 768 | 768 |
| 42 | 482.6933 | -768 | 256 | -256 | 768 | 768 | 1280 | -1280 | -768 | -768 | -768 |
| 43 | 511.7302 | -768 | 256 | -768 | -768 | -768 | -256 | -1280 | 1280 | 768 | 768 |
| 44 | 306.5272 | -768 | 256 | -768 | -768 | -768 | -1280 | 1280 | -1280 | -768 | -768 |
| 45 | 468.8461 | -768 | -768 | 256 | -768 | 768 | 256 | -256 | -768 | -768 | 768 |
| 46 | 402.3995 | -768 | 768 | 256 | -768 | 768 | 1280 | 256 | 768 | 768 | -768 |
| 47 | 312.9472 | 768 | -768 | 768 | 768 | -768 | -256 | 256 | -1280 | -768 | 768 |
| 48 | 205.9053 | -768 | -768 | 768 | 768 | -768 | -1280 | -256 | 1280 | 768 | -768 |
| 49 | 83.8950 | -768 | 768 | 768 | -768 | -1280 | 256 | 1280 | 768 | -768 | -768 |
| 50 | 27.7485 | -768 | 768 | 768 | -768 | -1280 | 1280 | -1280 | -768 | 768 | 768 |
| 51 | 139.4606 | -768 | 768 | 256 | 768 | 1280 | -256 | -1280 | 1280 | -768 | -768 |
| 52 | 239.3124 | -768 | 768 | 256 | 768 | 1280 | -1280 | 1280 | -1280 | 768 | 768 |
| 53 | 318.4214 | -768 | -256 | -768 | 768 | -1280 | 256 | -256 | -768 | 768 | -768 |
| 54 | 375.2791 | -768 | -256 | -768 | 768 | -1280 | 1280 | 256 | 768 | -768 | 768 |
| 55 | 435.[illegible] | -768 | -256 | -256 | -768 | 1280 | -256 | 256 | -1280 | 768 | -768 |
| 56 | 409.41[illegible] | -768 | -256 | -256 | -768 | 1280 | -1280 | -256 | 1280 | -768 | 768 |
| 57 | 386.1[illegible] | 768 | -768 | -768 | -768 | -768 | 1280 | -1280 | 1280 | 768 | -256 |
| 58 | 338.3148 | 768 | -768 | 768 | -768 | 768 | 256 | 1280 | -1280 | -768 | 256 |

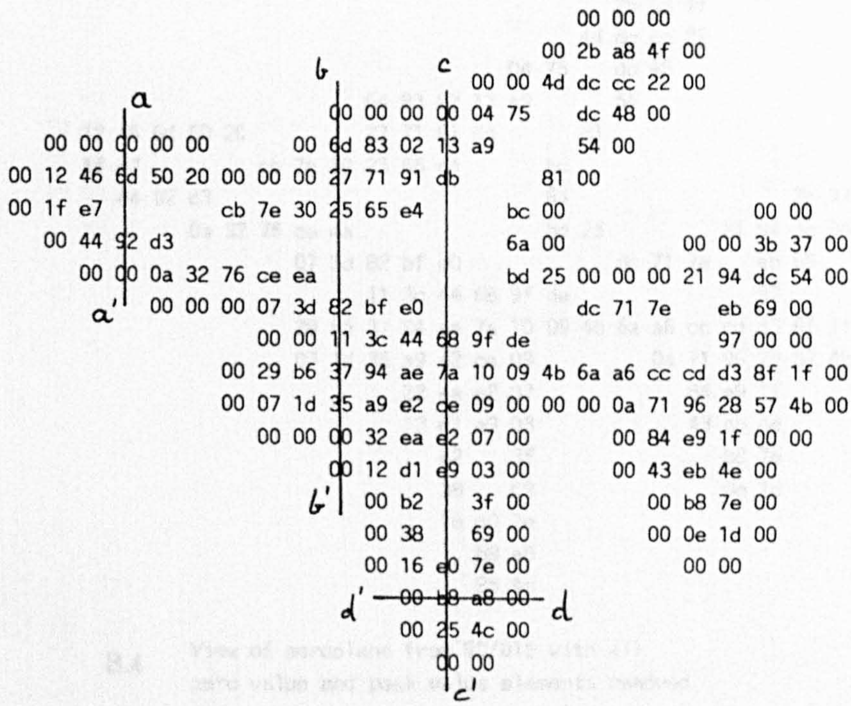
10. Comparison of Fourier and Walsh transforms

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 2b a8 4f 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4d dc cc 22 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 75 ec dc 48 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6d 83 02 13 a9 ec ec 54 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 12 46 6d 50 20 00 00 00 27 71 91 db ec ec 81 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 1f e7 ec ec ec cb 7e 30 25 65 e4 ec ec bc 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 44 92 d3 ec ec ec ec ec ec ec ec ec 6a 00 00 00 00 00 00 3b 37 00 00 00
00 00 00 00 00 00 00 00 00 00 0a 32 76 ce ea ec ec ec ec ec bd 25 00 00 00 21 94 dc 54 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 07 3d 82 bf e0 ec ec ec ec dc 71 7e ec eb 69 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 11 3c 44 68 9f de ec ec ec ec ec 97 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 29 b6 37 94 ae 7a 10 09 4b 6a a6 cc cd d3 8f 1f 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 07 1d 35 a9 e2 ce 09 00 00 00 0a 71 96 28 57 4b 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 32 ea e2 07 00 00 00 00 84 e9 1f 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 12 d1 e9 03 00 00 00 00 43 eb 4e 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 b2 ec 3f 00 00 00 00 00 b8 7e 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 38 ec 69 00 00 00 00 00 0e 1d 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 e0 7e 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 b8 a8 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 25 4c 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

**B.1** Digitised data points for view of aeroplane 90/015

```
                                                00 00 00
                                             00 2b a8 4f 00
                                       00 00 4d dc cc 22 00
                           00 00 00 00 04 75 ec dc 48 00
   00 00 00 00 00          00 6d 83 02 13 a9 ec ec 54 00
00 12 46 6d 50 20 00 00 00 27 71 91 db ec ec 81 00
00 1f e7 ec ec ec cb 7e 30 25 65 e4 ec ec bc 00                00 00
   00 44 92 d3 ec ec ec ec ec ec ec ec ec 6a 00          00 00 3b 37 00
      00 00 0a 32 76 ce ea ec ec ec ec ec bd 25 00 00 00 21 94 dc 54 00
            00 00 00 07 3d 82 bf e0 ec ec ec ec dc 71 7e ec eb 69 00
                     00 00 11 3c 44 68 9f de ec ec ec ec ec 97 00 00
                  00 29 b6 37 94 ae 7a 10 09 4b 6a a6 cc cd d3 8f 1f 00
                  00 07 1d 35 a9 e2 ce 09 00 00 00 0a 71 96 28 57 4b 00
                     00 00 00 32 ea e2 07 00       00 84 e9 1f 00 00
                           00 12 d1 e9 03 00       00 43 eb 4e 00
                              00 b2 ec 3f 00          00 b8 7e 00
                              00 38 ec 69 00          00 0e 1d 00
                              00 16 e0 7e 00             00 00
                                 00 b8 a8 00
                                 00 25 4c 00
                                    00 00
```

**B.2** View of aeroplane after removing zero values that carry no additional information



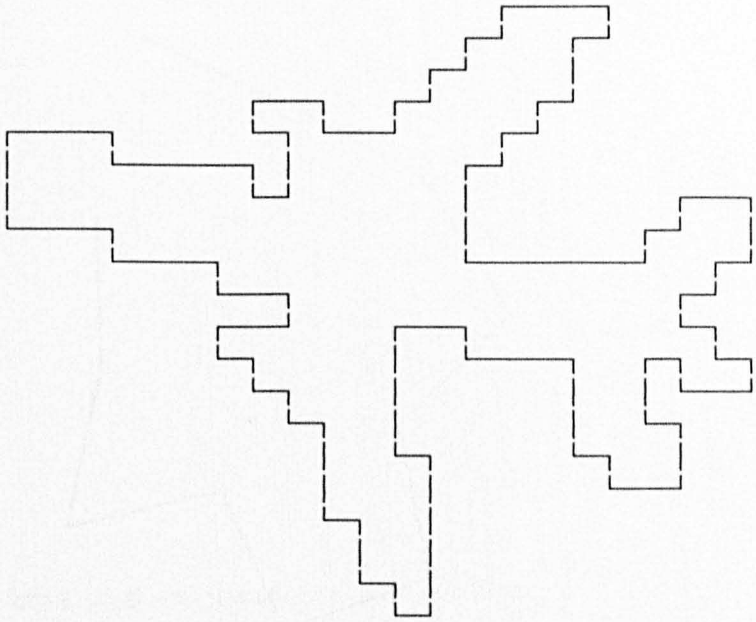

**B.3** View of aeroplane after removing zero and peak values that carry no additional information

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | 2b | a8 | 4f | | | | |
| | | | | | | | | | | | | | | 4d | dc | cc | 22 | | | | |
| | | | | | | | | | | | | 04 | 75 | | dc | 48 | | | | | |
| | | | | | | | | 6d | 83 | 02 | 13 | a9 | | | 54 | | | | | | |
| 12 | 46 | 6d | 50 | 20 | | | | 27 | 71 | 91 | db | | | 81 | | | | | | | |
| 1f | e7 | | | | cb | 7e | 30 | 25 | 65 | e4 | | | bc | | | | | | | | |
| | 44 | 92 | d3 | | | | | | | | | | 6a | | | | | | | 3b | 37 |
| | | | 0a | 32 | 76 | ce | ea | | | | | | bd | 25 | | | | 21 | 94 | dc | 54 |
| | | | | | | 07 | 3d | 82 | bf | e0 | | | | | dc | 71 | 7e | | eb | 69 | |
| | | | | | | | | 11 | 3c | 44 | 68 | 9f | de | | | | | | 97 | | |
| | | | | | | 29 | b6 | 37 | 94 | ae | 7a | 10 | 09 | 4b | 6a | a6 | cc | cd | d3 | 8f | 1f |
| | | | | | | 07 | 1d | 35 | a9 | e2 | ce | 09 | | | | 0a | 71 | 96 | 28 | 57 | 4b |
| | | | | | | | | | 32 | ea | e2 | 07 | | | | | 84 | e9 | 1f | | |
| | | | | | | | | | 12 | d1 | e9 | 03 | | | | | 43 | eb | 4e | | |
| | | | | | | | | | | b2 | | 3f | | | | | | b8 | 7e | | |
| | | | | | | | | | | 38 | | 69 | | | | | | 0e | 1d | | |
| | | | | | | | | | | 16 | e0 | 7e | | | | | | | | | |
| | | | | | | | | | | | b8 | a8 | | | | | | | | | |
| | | | | | | | | | | | 25 | 4c | | | | | | | | | |

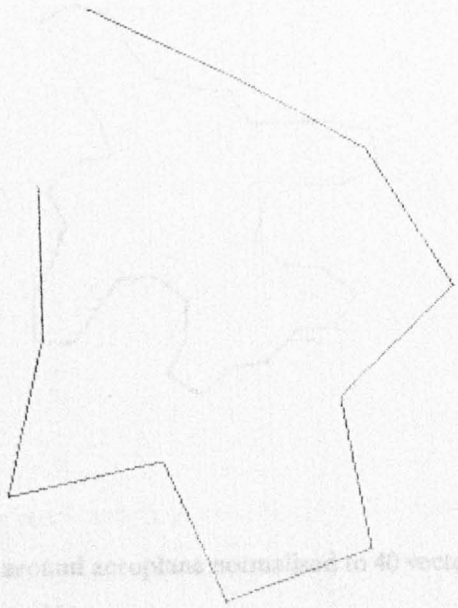
**B.4** View of aeroplane from 90/015 with all zero value and peak value elements removed

```
                                          00 2b a8 4f 00
                                       00 4d    cc 22
                        00 00       04 75       48 00
   00 00 00          00 6d 83 02 13 a9       54 00
12 46 6d 50 20 00 00 00 27 71 91 db       81 00
1f e7       ec cb 7e 30 25 65          bc 00                00 00
00 44 92 d3             ec             6a 00             00 3b 37 00
   00 00 0a 32 76 ce                   bd 25 00 00 00 21 94    54 00
            00 00 07 3d 82                ec dc 71 7e ec    69 00
                     00 11 3c       9f de                97 00
                  29 b6 37       7a 10 09 4b 6a a6       d3 8f 1f
                     1d 35       ce 09    00 00 0a 71 96 28 57 4b 00
                        00 32    e2 07          00 84 e9 1f 00 00
                           12 d1 e9 03          00 43    4e 00
                           00 b2    3f 00          00 b8 7e 00
                           00 38    69 00             0e 1d
                              16 e0 7e 00
                              00 b8 a8 00
                                 25 4c 00
                                    00
```

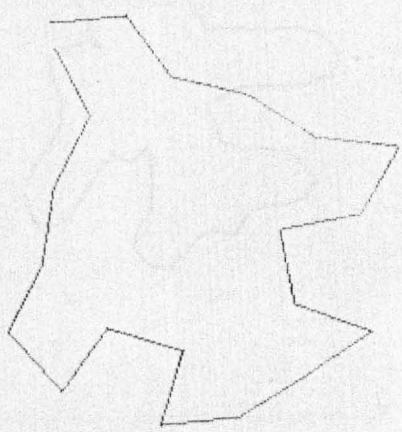
**B.5** Data values used in analysis of view 90/015



B.6 Coarse segmentation produced from data in view 90/015

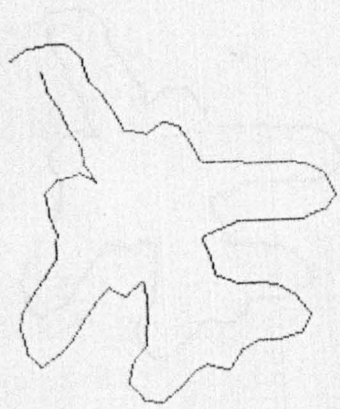


1. Contour around aeroplane normalised to 10 vectors



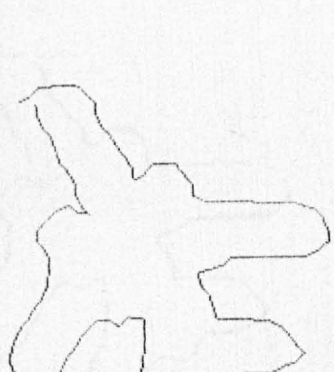
2. Contour around aeroplane normalised to 20 vectors

**3. Contour around aeroplane normalised to 40 vectors**

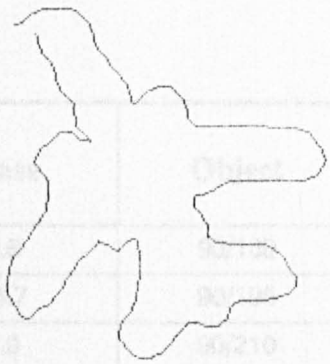


**4. Contour around aeroplane normalised to 75 vectors**

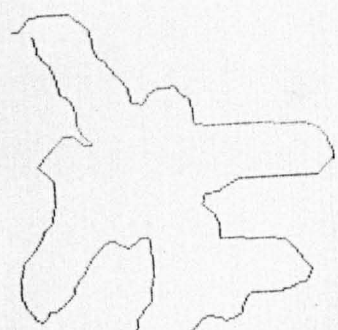
5. Contour around aeroplane normalised to 100 vectors



6. Contour around aeroplane normalised to 125 vectors



7. Contour around aeroplane normalised to 150 vectors



8. Contour around aeroplane normalised to 200 vectors

| Object | Phase | Object | Phase |
|---|---|---|---|
| 90/000 | -2.6 | 90/180 | 13.0 |
| 90/015 | -16.7 | 90/195 | 0.3 |
| 90/030 | -3.0 | 90/210 | 0.2 |
| 90/045 | -0.5 | 90/225 | 9.7 |
| 90/060 | -9.0 | 90/240 | 19.6 |
| 90/075 | -9.4 | 90/255 | 1.7 |
| 90/090 | -6.0 | 90/270 | -9.3 |
| 90/105 | 15.0 | 90/285 | 25.2 |
| 90/120 | 6.7 | 90/300 | -3.1 |
| 90/135 | 14.6 | 90/315 | 18.4 |
| 90/150 | 26.2 | 90/330 | 18.3 |
| 90/165 | 30.9 | 90/345 | 0.3 |