Image Segmentation

using

Local Surface Fitting

Adrian Wright

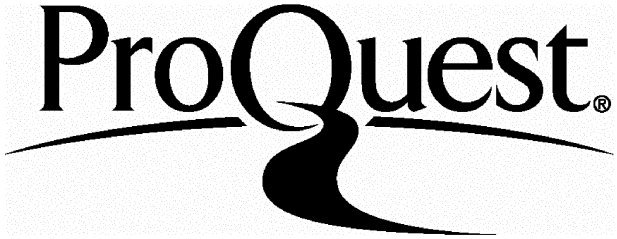
University College London

Thesis presented for the Degree of

Doctor of Philosophy

in the

University of London



Abstract:

Images contain information and the aim of digital image processing is generally to make the extraction of this information easier or even to automate it. Segmentation is the division of digital images into regions. The final aim is usually *scene* segmentation in which the regions correspond to actual objects in the image, for example a car, a brain tumour, a flooded area.

A more fundamental process is *image* segmentation, the subject of this thesis, which is entirely context free and results in regions which are homogeneous in themselves but may not necessarily correspond to whole objects. The basic assumption here is that all images can be segmented into regions that have certain consistent characteristics. A region classification is defined which postulates two basic region features, a 'smooth' grey level variation overlayed by textural variations which include 'noise'. This thesis is concerned with the 'smooth' grey level variations and the problems they present for segmentation.

A novel method of calculating the degree of connectivity of neighbouring pixels is presented based on the similarity of their best fitting local surfaces. The strength of a segmentation is defined as a sum, taken over all pixel-neighbour pairs, of these connectivities. The 'best' segmentation is defined as that with the greatest strength and it is shown how this definition parallels the intuitive idea of a good segmentation by completing boundaries even where they are locally weak.

Methods of approaching this defined 'best' segmentation are described starting with a simple thresholding of the connectivities and proceeding to optimisation using simulated annealing. Results using simulated and real images are presented and informally compared with a widely used segmentation technique based on Markov Random Fields.

Acknowledgements:

I was lucky enough to have two superb supervisors who never failed to help when asked:

I thank Dr. Mark Hodgetts for his constant encouragement without which I probably would not have got this far.

I thank Dr. Terry Fountain for his kindness, courtesy and support.

I also thank Dr. David Crawley for giving me access to his apparently inexhaustible supply of knowledge of computing software and hardware and for his generous patient assistance.

Contents:

List of Figures:

List of Tables:

1. Introduction - Images and Segmentation

1.1 Overview and Motivation

Images contain information and often the purpose in examining them is to extract a particular subset of this information to enable a task to be performed. The aim of digital image processing is generally to make this extraction of information easier or even to automate it; either partially, perhaps by reducing a large number of images to a more manageable data set for human examination, or totally, so that the task need no longer involve human operators.

There is a great variety of image types and of the tasks that they are used for: detecting and recognising a ship in an image of the sea; locating and reading the post-code on an envelope; finding a car in an image taken over a motorway, locating the registration plate and reading it; comparing a satellite image with a known map of country boundaries and identifying cloud formations; locating a cell on a microscope image of tissue and delineating the nucleus; recognising and measuring the gap between two metal plates to control a robot welder; finding a tumour in a MRI brain scan and determining if there is any change since a previous scan.

The previous examples show that often the requirement is to find a known type of object (ship, cell, car) in an image and measure some characteristic attribute connected with it. This process of object identification depends on recognising the required object(s) by making a comparison with a known object model and this implies that an object model must exist.

Occasionally this model might be fairly simple: blobs which are, say, round, big and bright could be found by searching the image directly using an adjustable template of the required object and registering the positions of highest correspondence.

This process is depicted in Figure 1-1.

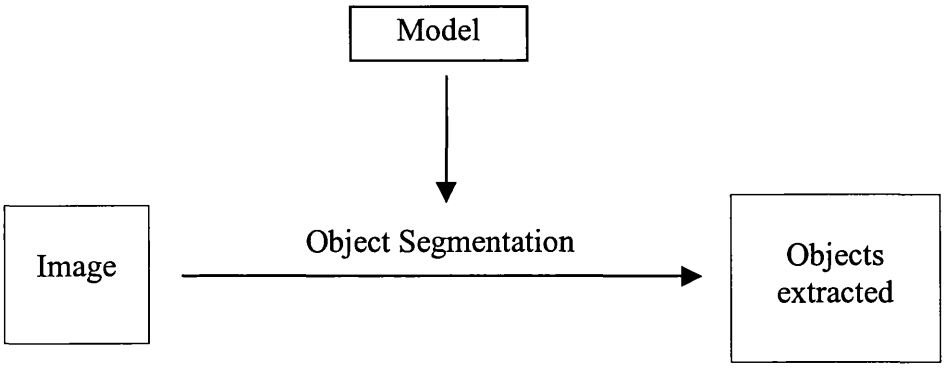


Figure 1-1 Object identification

However for many practical objects the model will be very complex; cars, cells, etc. are entities composed of a complicated arrangement of a, possibly large, number of simpler units. This complexity is increased by the fact that objects are often three-dimensional and the image is only one of many possible two-dimensional views and as a consequence the model needs to include the viewing arrangement and the different parameters associated with it. Attempting to apply such complex models directly to the basic image data is usually impractical; predicting the pixel grey levels for a possible object template, moving this over the whole image to discover any correspondences and repeating this for a possibly enormous number of different model parameter variations would certainly be too computationally time consuming for most applications. Furthermore, when the model consists of a total number of basic units and their relationships, which is small compared to the number of pixels in the image, it makes sense to reduce the image to the same type of units and then make comparisons of the model prediction with these.

These basic units will often be defined as homogeneous regions, uniform in grey level, texture or some other parameter, which are perhaps of a specific shape and orientation and topological relationship to each other. Locating these base regions is therefore the first task. This can be done by splitting the whole image into such homogeneous regions and then searching for those which satisfy the shape and orientation requirements. This process of delineating the uniform regions to form a labelled image in which each pixel is assigned to a specific region is termed segmentation. Following segmentation, the objects can be recognised by a pattern recognition procedure as depicted in figure 1.2.

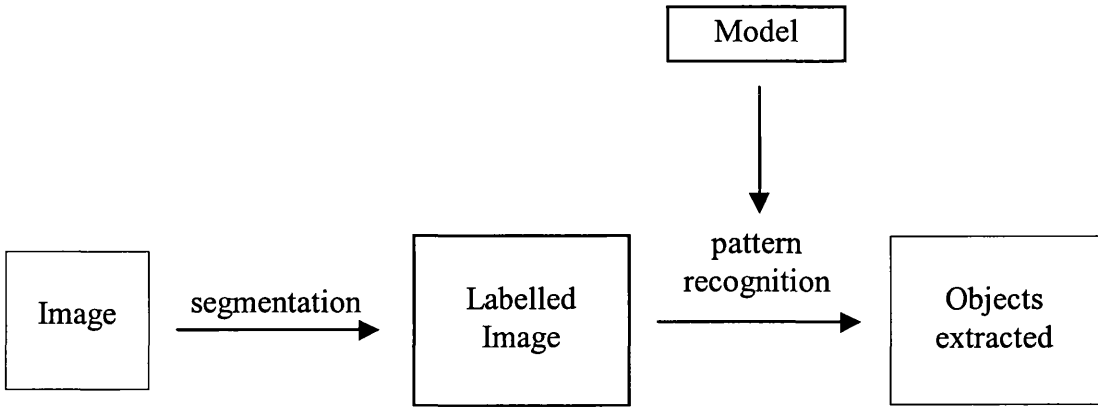
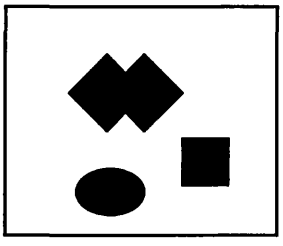
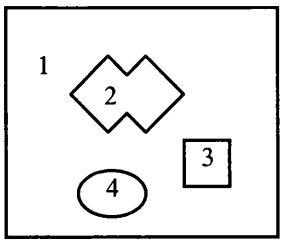


Figure 1-2 Scene segmentation

The complete process shown in figure 1.2 is defined by Nevatia[1] as 'scene segmentation'; the separation of the components of an image into subsets that correspond to the physical objects in the scene. The first part producing a labelled image identifying homogeneous regions he called 'image segmentation' and this simpler, model independent, process is often what is meant when the term segmentation is used on its own. Pratt[2] also adopted the same restricted definition of segmentation which does not require any contextual information or involve classifying the segments that are found. This definition is adopted here and it is important to distinguish this meaning from one that might require the use of any sort of explicit knowledge base of the objects that are required to be detected in the final analysis. It is especially important to recognise this distinction in making any evaluation of the worth of a particular segmentation scheme: an image segmentation will not necessarily produce regions corresponding to the objects of the scene. An example of this is given by consideration of the following simple binary toned image.



An *image segmentation* of this, using, for example thresholding or edge detection techniques, would easily produce the following regions in a labelled segmentation:



These four regions (number one being the background) could then be searched for any required property. If, for instance, square regions are of interest, then examining all regions for the property of having a perimeter consisting of four equal length straight sections joined at right angles would reveal just one region; number three, satisfying this condition (note that the boundary of the background region, number one, includes its borders with the other three regions and hence does not satisfy this requirement). Clearly in *scene segmentation* terms the omission of region two, which might well be caused by the overlap of two squares in the real object, is not satisfactory. In order to complete the scene segmentation from this starting point a model of the possible squares (size, colour, orientation, overlap possibilities) and some pattern recognition techniques which allow *all* the regions to be searched for 'squareness', are needed, as indicated in Figure 1-2. This example illustrates an important distinction that needs to be realised between merely checking existing regions for a particular property (for example squareness) and the much more difficult task of using this property to complete a scene segmentation from an existing image segmentation.

Real objects also might not be found by an image segmentation scheme for reasons other than overlap; a common difficulty, which will be examined later, is caused by the lack of a complete clear boundary between two or more objects. This might be the result of a gradual merging of the objects' grey level intensities at one or more points along their common boundaries.

The important point is that, even in seemingly very simple cases, scene objects will often not be found by image segmentation techniques and usually this can only be done with additional information about the required objects. Given this simple observation it does seem rather remarkable that many authors, having created a new image segmentation scheme which uses no object information whatsoever, then apply it to a real and often very complex image and attempt to evaluate its effectiveness in terms of

the real objects present in the image. As, according to the above observation, it might not be merely difficult but actually theoretically impossible to completely delineate such objects with such a technique this does seem, if not absurd, then certainly misguided. Perhaps what mitigates such an irrationality is the fact that simulated images can always be produced to order so that a given method will work well with them whereas real imagery does not allow of such artifice - although with such an abundance of real imagery available the biased selection of an artificially flattering image set is clearly possible.

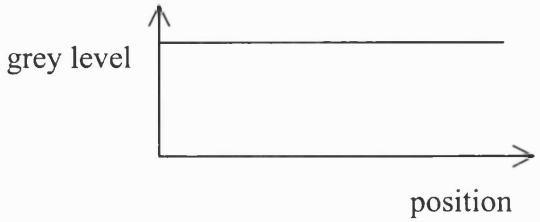
The quality of segmentation will improve with the amount of information that is available. In particular, better results may be obtained with multi-band images where the sensor records at more than one wavelength; the most obvious example being colour images. Multi-temporal images, a sequence of images taken at different times, are available from, for example, video cameras, and these are especially effective at enabling the isolation of moving objects against a constant background. Lastly, multiple images taken by sensors in different positions, provide information on the three-dimensional content of a scene. Combinations of these multi-image forming techniques can be also formed. Clearly, the most effective use of such multiple image sets will entail complex methods for fusing the information. Nonetheless, techniques for the segmentation of single images which are considered here, or modifications thereof, will still certainly still be of use.

This thesis will concentrate on 'image' segmentation: partitioning images into regions which are homogeneous independent of any contextual information To do this requires an understanding of the way that regions can be classified in a non-contextual way and this is dealt with in the next section.

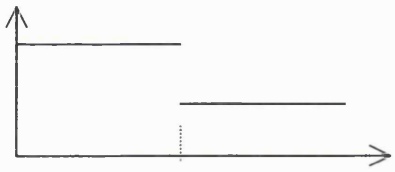
1.2 Region Classification

It is necessary to define the qualities that allow a region to be considered as a separate entity. By considering different types of images, starting at the simplest and gradually building up the complexity, it is hoped to show the changes that occur when new regions are formed. First the simplest possible structure is considered in which the grey level is constant over the whole image as shown below for a one dimensional image although the ideas extend naturally to two dimensions. Here, as in all of the following

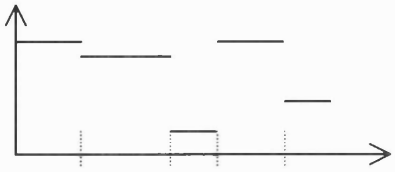
figures, the grey level is represented by the vertical axis and the pixel position by the horizontal axis.



It would seem evident that this image should sensibly be segmented into a single region. The next higher degree of complexity would be with two parts of the image each with constant but different grey levels:



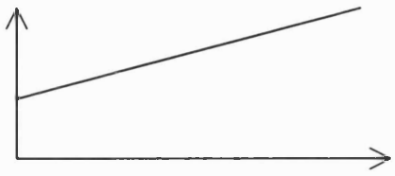
Again the optimum segmentation seems obvious; two regions separated at the junction of the change in grey level as indicated by the dashed line. This is easily extrapolated to include a greater number of grey level changes:



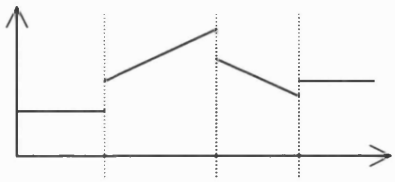
... and the regions are simply those areas with the same grey level. This model could be used to define the segmentation of all images; accumulate groups of pixels with the same grey level and label them as a individual regions. However this is unsatisfactory as there are regions of images with a variation in grey level which it would generally be wished to be considered as a single region and not split up into a possibly large collection of smaller regions. The types of variation in grey level which are possible

whilst still generally allowing the area to be delineated as a single region are next considered.

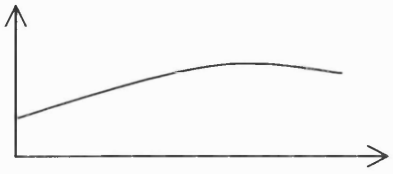
As a first example consider an image in which the grey level changes with a constant gradient:



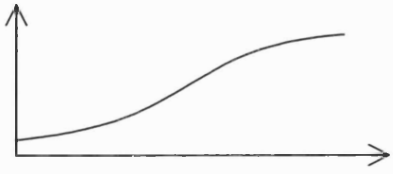
This smoothly changing grey level might occur, for example, if the visible lighting of an otherwise constant scene changed in a gradual manner due to the distance from the light source or if the reflectivity of the surface of an object changed due to its curvature or again, if the radiation measured depended on some physical property (density for example) which happened to be varying in a uniform way. In the absence of other features, such an area should be isolated as one region. And, as before, the segmentation of combinations of such region types is readily defined:



But the concept of the grey level changing in such a simple mathematical manner is clearly naïve and artificial; 'real' images can have areas which are considered as single regions but which have more complicated changes. What if the grey level changed 'smoothly' but not necessarily with a constant gradient, for example as in;

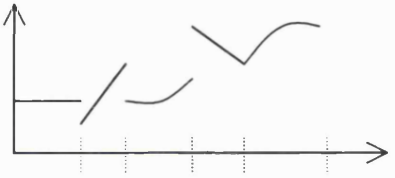


or

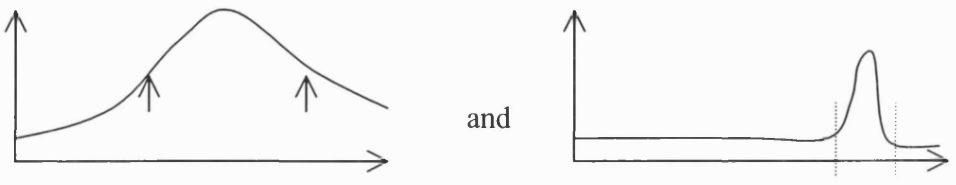


It would probably still be desirable to segment these images as single regions although now the description of the region would require more information. This would also be true for more extreme cases such as:

Where a description corresponding to the concepts of 'hill' or 'hole' would be necessitated. Again combinations of such regions produce images in which the segmentation is apparently well-defined:



However the issue of just when it is decided that a variation of this type is all one region and when it becomes more than one is not as clear as might at first be supposed. If two images are considered, each covered by a smooth variation in grey level as defined by the same function with different values of its parameters (for instance consider a Gaussian with different standard deviations):



Then, whereas the left hand image, as before, might reasonably be segmented into a single region it would seem preferable to split the right into three (in one dimension; two in two dimensions): a bright spot on a dark background with the boundaries somewhere as indicated by the dashed lines. It is interesting to consider why this difference in interpretation might be - after all if, as by definition, the two images are in one sense the same, being generated by the same function, then how can a different segmentation be required? One explanation is that, of course, normally regions are not characterised by what might be a rather arbitrary and complicated function. The normal requirement is to segment into regions with essentially *simple* characteristics: a slowly changing grey level gradient or, better, a constant grey level gradient or better still a constant grey level. These are the characteristics that are required of a *good* segmentation and indeed seem to be the way that the human eye-brain combination would divide up an image prior to the process of object identification.

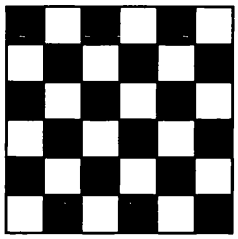
It is noted that the definition used by some edge detection operators - notably that of Marr - which are based on the detection of a zero crossing of the second derivative of the grey level (see Chapter 3 for a fuller description) would form boundaries in roughly the 'correct' place in the right-hand image above but would also segment the left-hand image into three regions at the points indicated by the arrows. This latter segmentation would seem to be, besides intuitively incorrect, also, in the context of later identification of objects associated with the regions, not entirely useful.

A second way in which the grey level can vary within an area which is then still accepted as a single region is in the case of texture. This is a non-uniform variation in grey level which consists of more abrupt spatial changes. Texture takes different forms and can be probabilistic or fully determined. At its simplest a change to each pixel's grey level can be made independently by the addition of a value from a known probability function of mean zero and specified variance. This would be a noise texture.

More complex probabilistic textures can have a spatial pixel interdependence as defined, for example by a Markov random field or the Grey Level Co-occurrence Matrix (see chap 3 for more details). These textures have a characteristic texture element(s) and a scale defined by the size of the texture element relative to the size of the region. Indeed it is the number of elements present that determine whether the

region is preferably described as a single textured region or as a collection of smaller, more uniform (untextured) regions. As a simple and therefore rather formal example of this consider the following image:

This would generally be considered as four regions of constant grey level. However by forming a simple repetition of the above the following is obtained.



And this, instead of being described as a 36 region segmentation would more conveniently be considered as a single region with a particular texture type. All textures have this property: if a sufficiently small area is taken then it becomes more effective to describe the result as a (small) number of uniform grey level regions (that is of the type described earlier - with a smoothly varying grey level).

This characteristic is perhaps the defining feature of the optimum segmentation: it condenses the image information to the minimum possible by describing the image as a set of regions each with its own description. The information includes both a description of the region boundaries and also the internal region characteristics. The individual region characteristic information can be reduced by choosing simple region types - the constant grey level regions described earlier - but this is at the expense of having a large number of regions (individual pixels in the case of noisy images) and this boundary information would become excessive. The other extreme would be to choose a relatively small number of regions with complicated internal features. The optimum segmentation is a balance somewhere between these two extremes.

To summarise, region features are of two types:

- a 'smooth' grey level variation

and

- textural variations

Note that:

1. All images can be divided into regions which only display a 'smooth' intensity variation.

2. The grouping of clusters of neighbouring such regions into single regions of 'texture' can be regarded as an additional segmentation stage which further condenses the image information description.

3. Textural variations have a large number of sub-divisions and these are not exclusive: different textural types can be superimposed. A common example would be of a simple noise texture overlaid on a more complex spatially dependent texture.

1.3 Preview of the remaining chapters

This thesis is concerned mainly with the 'smooth' grey level variations and the problems they present for segmentation. The overall structure is as follows:

- In chapter two a formal definition of segmentation is made and two methods of recording segmentations described.

- Chapter three contains a general summary of many segmentation methods that have been used and describes in some detail techniques which use a surface fitting method.

- Chapter four details a new method of calculating the degree of linking of neighbouring pixels dependent on the similarity of their associated best measured local surfaces and defines a formal 'best' segmentation in terms of these linkages.

- Chapter five shows how the linkages can be zero-thresholded to arrive at an approximate subset of the best segmentation or over-thresholded to obtain more regions.

- Chapter six demonstrates how the set of regions arrived at by over-thresholding can be optimised using simulated annealing combined with four-colour labelling to obtain the best subset of these regions.

- Chapter seven shows how the problem of undesired region merging caused by incomplete region boundaries can be partly resolved by dividing the image into smaller sub-images, applying the initial segmentation procedure independently within each sub-image and then optimising the final set of regions.

- Chapter eight contains an overall conclusion and recommendations for future investigations

2. Segmentation Definition, Recording and Counting

2.1 Segmentation Definition

Image segmentation requires the division of an image into regions (segments) each of which satisfies the criteria of *connectivity*.

The connectivity condition stipulates that for each pair of pixels in any region there must exist at least one path which passes only through neighbouring pixels which are also members of the same region. Neighbourhoods are defined as either four-way, in which only laterally adjacent pixels - up, down, left, right - are considered, or eight-way in which diagonally adjacent pixels are considered in addition. In this work, partly for convenience but also because of the use of four-colour labelling (see later) which is restrictive a confinement to 4-connectivity is made. Connectivity is a necessary condition to ensure that each region is a self-contained whole and not split into separate parts.

A more formal definition of a segmentation can be given as follows. Firstly let P be the infinite set of positive integers;

$$P \equiv \{1,2,3,4,5,.....\} \qquad \text{Eq 2-1}$$

and let the finite subset of P up to and including n be

$$P\#n \equiv \{1,2,3,4,5,.....n\} \qquad \text{Eq 2-2}$$

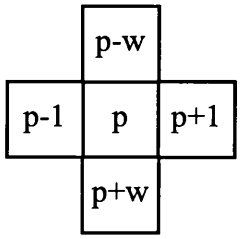
Now let an image, containing N pixels, numbered 1 to N, be segmented into a total of R, regions, numbered 1 to R, in which each pixel, p, is labelled with a region number, $l(p)$. As there are many such labellings it is appropriate to distinguish between them (see the next sections for a further description of labelling and estimates of numbers). This is done using a subscript; $l_a(p)$ is a particular labelling with a total of R_a regions. (Note that different labellings can have the same number of regions).

Then the labelling is actually a function;

$$l_a : P\#N \rightarrow P\#R_a \qquad \text{Eq 2-3}$$

Not all possible functions are segmentations; they need to satisfy the condition, given above, of connectivity.

It is possible to define connectivity as a restriction on the labelling as follows. First define the 4-connected neighbourhood, $Q_4(p)$ of any pixel, p. Assume the pixels to be numbered in order starting from the top left of the image and completing each horizontal row before moving to the next. If the image is of width w pixels then the numbers of the 4-connected neighbourhood of pixel, p are;



and therefore,

$$Q_4(p) = \{p-w, p-1, p+1, p+w\} \qquad \text{Eq 2-4}$$

with the proviso that these neighbourhoods will be incomplete for pixels which lie on the edge of the image. A region, r, part of the segmentation, l_a, is said to be connected, as stated before, if for every pair of pixels, u and v, say, which are members of the region, there exists at least one continuous path of neighbouring pixels all of which are also members of the same region.

The connectivity condition therefore can be expressed formally as:

$$l_a(u) = r = l_a(v) \Rightarrow \qquad \text{Eq 2-5}$$
$$\exists\{p_i, i = 1,..n \mid p_1 = u; p_n = v; \forall i : l_a(p_i) = r, p_{i+1} \in Q_4(p_i)\}$$

The connectivity condition is an absolute formal requirement for a segmentation. However many labellings satisfying this condition would not represent satisfactory

segmentations. In addition, the pixels of each region need to have some property of uniformity or similarity which is the reason why they are grouped together and there needs to be a reason for separating neighbouring regions.

A common method of specifying these requirements is with a homogeneity condition. This assumes that there exists a test, H, of any group of connected pixels, x, based on some calculable property which allows the group to be classified as either homogeneous (H(x)=True) or not homogeneous (H(x)=False). All regions should individually pass the test and all pairs of neighbouring regions taken together should fail.

The homogeneity condition could be (and often is) as simple as all the pixels having to have a similar grey level. A binary thresholding method automatically produces a segmentation satisfying this requirement by forcing pixels into one of two classes dependent only on their intensity. Split and Merge methods (see Chapter 3) also work directly with a homogeneity criterion to decide on the division or joining of regions.

However many segmentation methods do not rely on a region homogeneity definition but produce their results by different criteria. Furthermore a homogeneity condition is only a way of deciding on an acceptable segmentation and does not necessarily indicate a way of arriving at the optimum segmentation.

2.2 Methods of Recording Segmentations

Segmentations can be recorded in a number of different ways all of which are based on two main ideas: *labelling* and *linking*.

Labelling, as has already been described (see the previous section), is the process of assigning to each pixel a label which is the number of the region to which it belongs.

Labellings of this type are not unique; given R_a regions there are $R_a!$ different labellings; two examples for the same eight region segmentation of the same image are shown in Figure 2-1.

1	1	2	2	2	2
2	2	2	3	3	6
2	2	2	3	3	6
4	4	4	3	7	6
4	5	4	3	7	7
4	4	4	3	7	8

3	3	1	1	1	1
1	1	1	8	8	6
1	1	1	8	8	6
7	7	7	8	4	6
7	5	7	8	4	4
7	7	7	8	4	2

Figure 2-1 Equivalent labellings of an eight region segmentation

However, these different equivalent labellings are of no real interest and the only important consideration is to ensure that only one is chosen and then adhered to.

The labelling method described so far has used a total number of labels equal to the number of different regions, which conveniently gives each region a unique label. This is not necessary if the only purpose is to distinguish neighbouring regions from each other. The four-colour map theorem, Appel and Haken[3] states that just four colours or labels are necessary to ensure that neighbouring regions always have a different label. This is illustrated in Figure 2-2 in which the four labels, A,B,C,D, are used to colour the previous example from Figure 2-1. Four, then, is the maximum number of labels

1	1	2	2	2	2
2	2	2	3	3	6
2	2	2	3	3	6
4	4	4	3	7	6
4	5	4	3	7	7
4	4	4	3	7	8

A	A	B	B	B	B
B	B	B	D	D	C
B	B	B	D	D	C
C	C	C	D	B	C
C	A	C	D	B	B
C	C	C	D	B	A

Figure 2-2, Four colour labelling of an eight region segmentation

which is ever necessary to describe any segmentation, although in some simple cases even four are not always needed - the example in Figure 2-2 can, in fact, be achieved with only three labels (for example, the single region labelled 'D' could be relabelled 'A').

Although only four labels are needed it is possible to use more; in fact for any segmentation any number of labels between four and R_a inclusive can be used. This concept is used further in Chapter 6. Note that, although there is a simple procedure for converting a four-colour labelling to the full region-number labelling by finding each connected four-colour labelled region in turn and relabelling its pixels to the next available region number, there is no such simple formal procedure for the reverse process: if there were, the four-colour theorem would not have been so difficult to prove!

Linking, the second general method of recording a segmentation, is based on recording the existence or otherwise of links between a pixel and each of its neighbours. A link is taken to mean that the pixel and its neighbour are part of the same region. Again, either 4 or 8 neighbourhoods can be considered but here concentration is made on 4-connections. Define L(p,n) as being the link between a pixel, p, and one of its neighbours, n, with a value of one indicating that they are linked and a value of zero meaning that they are not. Clearly from symmetry considerations;

$$L(p,n) = L(n,p) \qquad \text{Eq 2-6}$$

and this means that from a practical point of view it is only necessary to store two of the four links at each pixel position, the other two being stored at the neighbouring positions.

Links and labels are not in a simple one to one correspondence with each other. However a set of links can easily be used to produce a labelled image by the relatively simple procedure of always giving neighbouring pixels the same label if they are linked;

$$L(p,n) = 1 \;\Rightarrow\; l(p) = l(n) \qquad \text{Eq 2-7}$$

The pixels of all regions in turn can be labelled starting from a currently unlabelled pixel, giving it the next available region label and then giving the same label to those of its linked neighbours followed by their linked neighbours and so on until no more are accumulated. In this way a labelling can be obtained from a set of links,

Eq 2-8

$$L(p,n) \Rightarrow l(p)$$

Any two pixels, v and w, say, will be given the same label providing only that there exists at least one continuous path of neighbouring pixels connecting them, each of which is linked to the next. It is important to realise that, although this method of obtaining labels from links seems obvious, it is by no means unique; other, equally valid ways of obtaining a labelling, are possible and, indeed, this simple method has associated problems which are described here and in later chapters.

The relationship, Eq 2-8 , is not necessarily reversible; different sets of links can produce the same labels. Figure 2-3 shows three different link sets all producing the same labels.

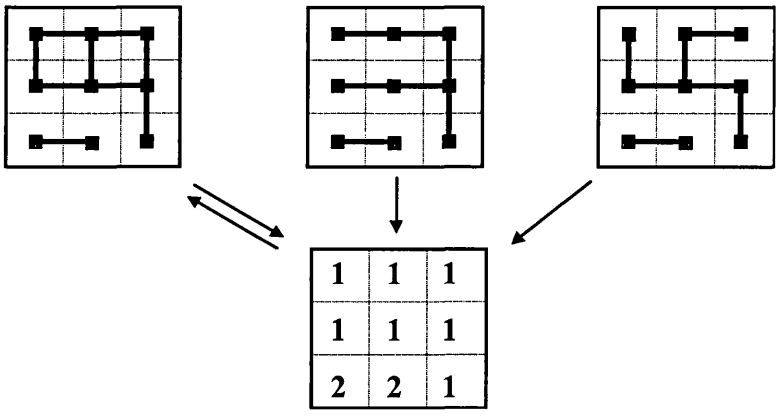


Figure 2-3 Equivalent link sets

This is because some of these sets contain inconsistencies; pairs of neighbouring pixels which are not directly linked are joined by an indirect route resulting, by the definition, Eq 2-7 , in the same labelling. Any chain of non-links which is not closed is inevitably redundant when the labelling process is applied; for example the fragment of image in Figure 2-4 which contains a chain of four non-links, will nonetheless result in all the pixels having the same label.

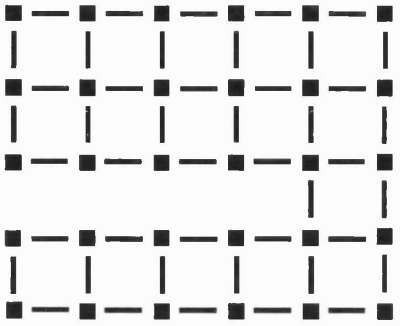


Figure 2-4 Link set with redundant non-links

Inconsistencies in link sets can be defined and hence detected, in terms of the arrangement of links at each four pixel square neighbourhood. Of the six possible arrangements with zero, one, two, three, or four links present, the five allowable, self-consistent arrangements are depicted in Figure 2-5 which also shows the resulting region boundaries for each arrangement.

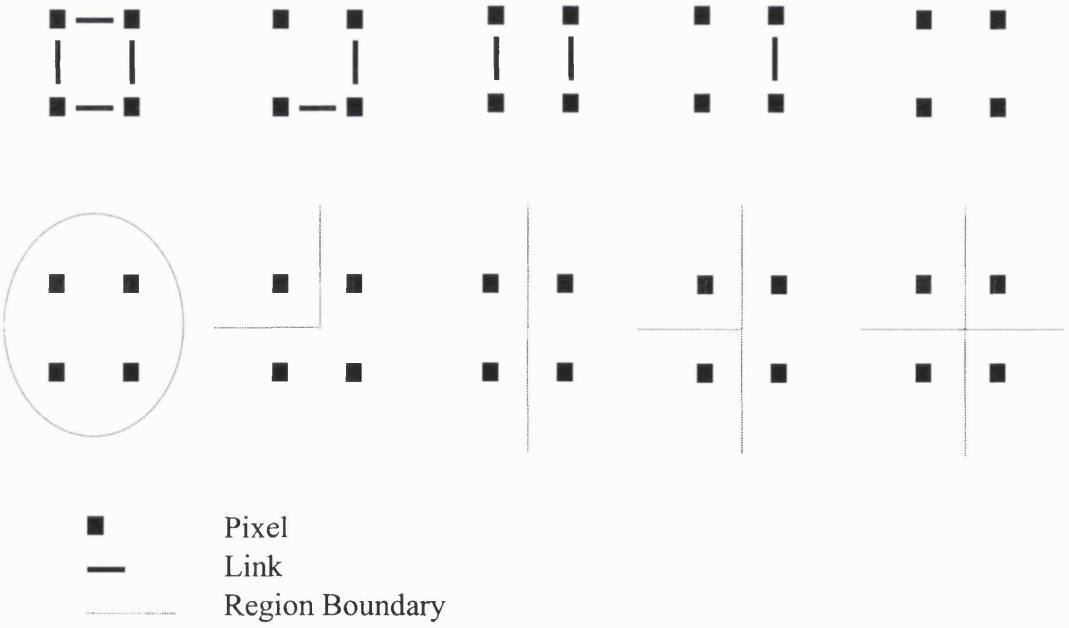
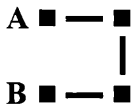


Figure 2-5 Self-consistent link arrangements in four-pixel neighbourhoods

The remaining possible arrangement of links in a four pixel neighbourhood;



is not self-consistent, pixels A and B being both linked by an indirect route and also directly not-linked. Therefore the presence of just one four pixel neighbourhood with three links and one non-link is an indication of an inconsistent link set.

What might be termed the maximal consistent set of links is obtainable from a given labelling by the process;

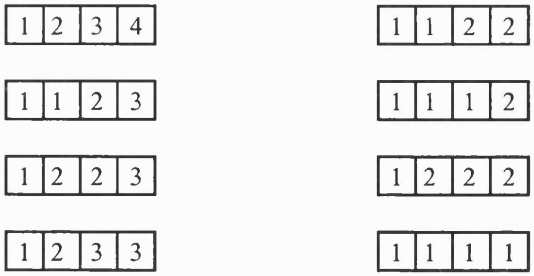
$$l_a(p) = l_a(n) \Rightarrow L_a(p,n) = 1 \qquad \text{Eq 2-9}$$
$$l_a(p) \neq l_a(n) \Rightarrow L_a(p,n) = 0$$

In Figure 2-3 this set is the leftmost of the three shown.

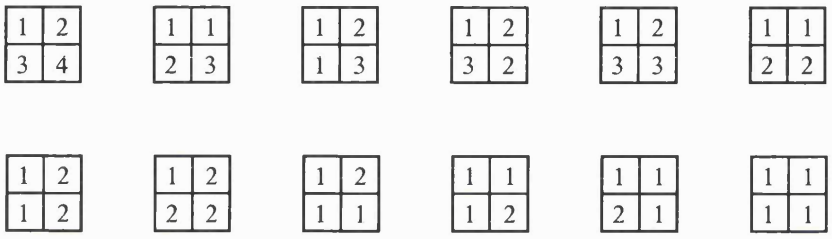
Link sets are closely related to edge detected images in which pixels are designated as parts of an edge - *edgels* - if their gradient with neighbouring pixels is above some prescribed limit. Storing the gradient in two directions is then equivalent to a link set with links of one when the gradient is less than the limit and zero when it is above. Edge images suffer from the same problem as link sets in that an incomplete chain of edgels does not result in a region boundary.

2.3 Combinatorial Analysis

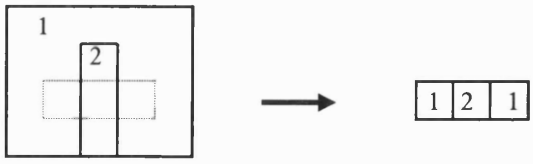
Counting the number of different possible segmentations for a given image is a non-trivial problem in itself. First consider gradually building an image consisting of a single row of pixels by the process of adding one pixel to the end of the row at each step. For the region number of each new pixel there are always two choices; it can be set to the same as that of the last pixel or a completely new number can be used (apart from the first pixel which can always be set to one without loss of generality). It is not allowed to apply any other already used region number as this would result in an unconnected region. This means that for *a row image* of N pixels there are 2^{N-1} possible segmentations. For a 4-pixel row image this gives a total of $2^3 = 8$ segmentations and these are shown below:



However if the same number of pixels are arranged differently then in general rather more possible segmentations will be obtained. For example, still with just four pixels but now arranged as a 2 by 2 image, there are 12 segmentations:



Counting the number of possible segmentations for other than row images is much more complicated. The image cannot be gradually constructed in the same way as the row case could be, because, although the final result must be a segmentation with connected regions, the intermediate stages need not be. This can be seen by considering the following two-region segmentation containing a section outlined by the dashed line which, if considered separately, contains a non-connected and hence illegitimate region:



Segmentations in general cannot therefore be either constructed from or chopped up into, smaller sub-segmentations. In fact, as there are different numbers of segmentations for the same numbers of pixels depending on their arrangement, clearly there can be no single formula which gives the number of segmentations purely in

terms of the number of pixels. The best that can be hoped for is to estimate bounds for the number of segmentations, S. The row image is the simplest with presumably the smallest number of possible segmentations and therefore using the previous calculation;

$$S \geq 2^{N-1}$$

An upper bound can be estimated by considering four-colour labellings. There are 4^N of these and each is mappable to one region number segmentation and from the four-colour theorem these must include all possible segmentations. However not all the resulting segmentations are different. For example there are four of the labellings which only use one label and these all result in the same segmentation which consists of just one region. Others use only two or three labels but clearly in a practical size of image consisting of a large number of pixels the great majority of labellings will use all four labels. Each of these will exist in 4! = 24, permutations which will result in the same segmentation. Therefore as an estimate of an upper bound;

$$S \lessapprox \frac{4^N}{4!}$$

It should be emphasised that this is only an estimate, firstly because the divisor 4! is only correct for those labellings using all four labels. Secondly, and more importantly, there exist four-colour labellings which are not simple permutations of each other, but which nevertheless result in the same segmentation; for instance it is always possible to change the label of any single region whose neighbouring regions only use one or two of the remaining three labels and obtain the same segmentation. Also, as previously shown, some labellings using all four labels are in fact equivalent to a three-labelling. Combining these upper and lower bounds;

$$2^{N-1} < S \lessapprox \frac{4^N}{4!}$$

or,

$$2^{N-1} < S \lessapprox \frac{2^{2N-3}}{3}$$

which in powers of 10 can be written;

$$10^{0.3*(N-1)} \lessapprox S \lessapprox \frac{10^{0.3*(2N-3)}}{3}$$

from which the following table can be constructed:

N, number of pixels	number segmentations, lower bound	number segmentations, upper bound
5	16	42
10	501	$4.2\ 10^{4}$
100	$5.0\ 10^{29}$	$4.2\ 10^{58}$

Table 2-1, Bounds on numbers of segmentations

As can be seen, the number of possible segmentations rapidly becomes extremely large even for relatively small images. This is of significance for the optimisation work in Chapters 6 and 7.

2.4 Summary

- A definition of segmentation was made in terms of the homogeneity and connectivity conditions.
- Two methods of recording segmentations were described: labelling and linking and their relationship examined.
- The number of possible segmentations for a given size of image was estimated

3. Segmentation Review

3.1 Segmentation in general

Many techniques now exist for the segmentation of images, and new methods are continually being developed. A 'BIDS' search for 1995 alone revealed 332 papers with the words 'image' and 'segmentation' in the title/abstract. Segmentation methods include such a variety of techniques that classifying them into useful groupings is a task that has led to contrasting results when attempted by different authors. This is illustrated by briefly summarising three segmentation reviews published between 1981 and 1993.

A review by Fu and Mui[50] published in 1981, divides methods into three main classes with a considerable hierarchy of sub-classes:

1. Characteristic feature thresholding or clustering
    1.1. Thresholding
        1.1.1. Statistical
        1.1.2. Structural
    1.2. Clustering
2. Edge detection
    2.1. Parallel techniques
        2.1.1. Edge element extraction
            2.1.1.1. High emphasis spatial frequency filtering
            2.1.1.2. Gradient operators
            2.1.1.3. Functional approximations
        2.1.2. Edge element combination
            2.1.2.1. Heuristic search and dynamic programming
            2.1.2.2. Relaxation
            2.1.2.3. Line and curve fitting
    2.2. Sequential techniques
3. Region extraction.
    3.1. Region merging
    3.2. Region dividing

3.3. Region merging and dividing

Publishing their survey in 1985 Haralick and Shapiro[53] initially define six different segmentation 'schemes':

1. measurement space guided spatial clustering,
2. single linkage region growing,
3. hybrid linkage region growing,
4. centroid linkage region growing,
5. spatial clustering
6. split-and-merge.

But later sections further sub-divide the first class (measurement space guided spatial clustering) to include two sub-classes: 'thresholding' and 'multi-dimensional measurement space clustering' and, somewhat confusingly, characterise classes 2 to 4 as being sub-classes of just one: 'region growing' (see later for a description). As, in addition, classes 1 and 5 seem to be strongly related (viz. 'spatial clustering') and class 6 'split and merge' includes a type of 'region growing' it is clear that, even for well-respected authors, classifying segmentation methods into well-defined and separated categories is a non-trivial task.

A review by Pal and Pal[51] made in 1993 rehashes the segmentation classes again, partly with a view, as the authors say, of including new developments. Their categories are:

1. Grey level thresholding
2. Iterative pixel classification, further sub-divided into:
   2.1. Relaxation
   2.2. Markov random field based approaches (see later for definition)
   2.3. Neural network based approaches
3. Surface Based Segmentation
4. Segmentation of Colour Images
5. Edge Detection
6. Methods Based on Fuzzy Logic Set Theory, divided into;
   6.1. Fuzzy Thresholding
   6.2. Fuzzy clustering
   6.3. Fuzzy edge detection

It is evident that one problem with defining a satisfactory classification of segmentation methods is that individual methods are often a combination of two or more techniques that by themselves do not constitute a complete segmentation scheme. Different combinations of these techniques produce different segmentation methods.

In the following, the major techniques that are in current usage are listed and some recent applications described. It is emphasised that several techniques are not complete segmentation methods in themselves and that some references, which use a multiplicity of techniques, will be quoted in more than one section.

- ***Thresholding***

Thresholding is the process of dividing pixels into different classes depending on where their grey value lies relative to a set of thresholds. Sahoo et al[54] survey thresholding methods, grouping them into categories of global, local and multithresholding. The simplest methods are bilevel global techniques which use the same single threshold over the whole image to divide the pixels into two levels. Pixels with a grey value less than the threshold are labelled as being in class 1 (say) and those with grey level values equal to or greater than the threshold are labelled as being in class 2. This does not necessarily result in two regions, one of class 1 pixels and the other of class 2 pixels, as the connectivity requirement still has to be satisfied (all pairs of pixels in the same region must be connected by a continuous chain of pixels also in the same region - see Chapter 2 for a more formal definition) and it is quite possible that not all the pixels of one class are all connected .

Many different techniques now exist for choosing a threshold. These are often partly based on knowledge of, or assumptions made about, the image content and its noise statistics. Teuber[72] derives the optimal threshold for a binary image assuming that the background and object levels are given by Poisson distributions. A recent paper by Hannah et al[5] uses methods based on variance and entropy to select thresholds for the detection of foreign objects in food packets. Brink[56] also uses entropy considerations to select a threshold for a bimodal histogram. Gupta and Sortrakul[23] develop a commonly used technique in which the probability density of the grey levels in the image is a mixture of two Gaussian density functions corresponding to 'targets' and the background.

Local methods have also been devised which do not use the same threshold over the whole image but allow it to vary depending on local neighbourhood statistics. Bilevel methods arise from the frequent desire to separate the image into just two parts; a foreground and a background. However many of these methods can and have been extended to use multiple thresholds and enable more complex image types to be segmented. Wang and Haralick[4] define a method for recursively selecting thresholds based on the histogram information obtained for edge type pixels. Derayer and Hode[59] define an automatic system for selecting multiple thresholds based on first identifying 'bi-points' off the diagonal of the co-occurrence matrix (see later for definition) which define neighbouring regions of different intensities and hence are a suitable position for a threshold. Tsai[71] defines a 'fast' thresholding selection procedure based on the common assumption of a multimodal histogram.

Although essentially simple in concept, thresholding methods have a major advantage in being able to guarantee the production of complete closed regions. However, there may be considerably more regions than desired, especially in the presence of noise.

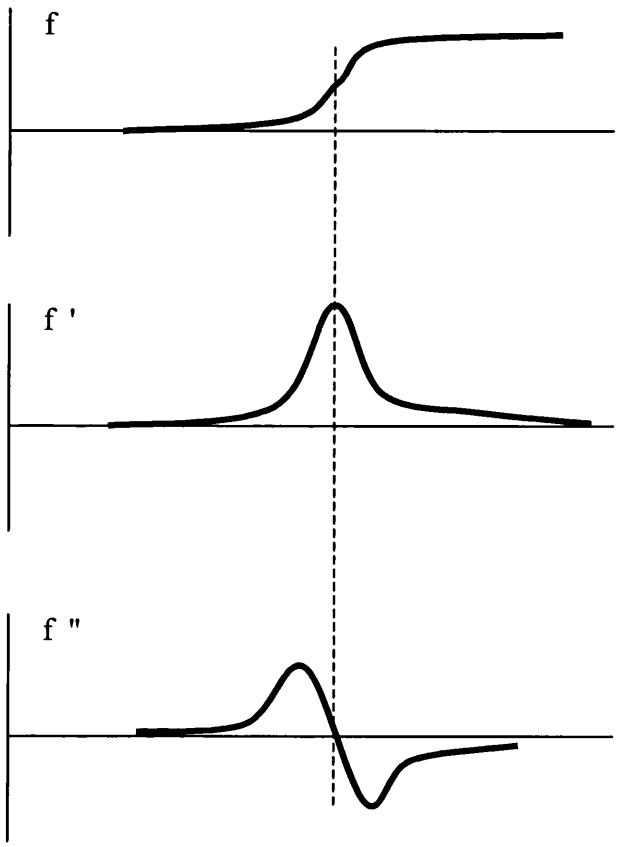
- ***Edge detection***

Edge detection processes find pixels that lie at the boundary of two regions. These pixels are, by definition, in a region where there is a change in grey level. Edge detectors examine each pixel neighbourhood and quantify the slope, and often also the direction, of the grey-level transition. An edge detection operation results in a binary image with pixels being labelled as either edge or non-edge. Regions can be formed from connected groups of non-edge pixels. The edge pixels must then be accounted for by assigning them to the most appropriate region.

Many edge detectors now exist, most of which are based on convolution with a set of directional derivative masks. The Roberts, Sobel, Prewitt and Kirsch operators (see for example Castleman[73]) all use small (e.g.; 3 by 3) window kernels to detect gradients in grey level. These simple operators all respond to noise as well as real edges but to some extent this can be counteracted by using some form of image smoothing as a preprocessing operation. A more complex operator is that of Canny[74] which finds the gradient maxima of a previously Gaussian smoothed image used at different scales. Le

Moigne and Tilton[40] use Canny edge detector features to refine region data produced from region growing.

The detector devised by Marr and Hildreth[55] is derived on a theoretical basis from assumptions about the image noise and uses the Laplacian operator to detect second rather than first derivatives in grey level structure. The following figure shows the first (f') and second (f'') derivatives of a simple smoothed step function in one dimension. It can be seen that the first derivative has a peak value where the gradient has a maximum and that this corresponds to a zero crossing in the second derivative.



Marr and Hildreth take these zero-crossings to define edges and in a two-dimensional image in which the edge orientation needs to be defined they take that with the maximum slope.

Haralick[9] uses a 'facet' model (see also section 3.2) to fit a polynomial to the image grey level data in small neighbourhoods. Edge pixels are then defined, similarly to

Marr and Hildreth, as those for which there is a negatively sloped zero crossing of the second directional derivative in the direction of non-zero gradient.

In a highly mathematical paper with formalised definitions, lemmas and theorems, Haddon and Boyce [62] propose a method of unifying region and edge information.

Iyengar and Weian[68] review edge detection methods and reject the commonly used smoothing pre-processing that is often used on the grounds that the inherent assumption of infinitely long edges is implausible and that corners need to be accounted for as well. They apply a relaxation technique to a combined Markov Random Field and Bayesian based model (see later for descriptions).

Edge detectors suffer from two general problems. Firstly they sometimes produce thick edges which require subsequent thinning to progress to segmentation. Secondly the edges produced rarely form complete closed boundaries and hence do not result in regions without a further edge-linking operation. Farag and Delp[41] describe an edge linking method based on a sequential search of an edge graph.

- ***Region Growing***

Region growing, as the name implies, is a technique in which an initially small region is increased in size by the accumulation of neighbouring pixels or other regions. Additions are only made if the homogeneity of the resulting region, as measured by some parameter such as grey level variation, is within some set limits. Starting with the whole image divided into small regions or simply with the original pixels, the process is iterated until no further combinations can be made and a segmentation obtained. Haralick and Shapiro[53] review region growing schemes in some depth, defining three classes including 'single linkage' methods, in which neighbouring pixels are compared for merging using the values of two pixels only. These schemes have the virtue of simplicity but are prone to allowing unwanted merging of regions due to a chain caused by just one pair of neighbouring similar pixels across a boundary. 'Hybrid single linkage' schemes are more robust, as property vectors based on the local neighbourhood of pixels are used. 'Centroid linkage region growing' schemes compare pixels, not just to their neighbours, but to existing growing whole regions.

Pong et al[17] describe a region growing scheme based on the facet model of Haralick. Initial 'atomic' regions are constructed and then the construction of a region adjacency graph and a region property list enables the merging of regions to be undertaken.

Spann and Horne[14] use a pyramidal hierarchy of regions applying clustering at each level. Choo et al[58] construct a system for growing regions from a starting block size which is suitable for complex, highly textured scenes.

Adams and Bischof[13] describe a seeded region growing method which requires the input of a number of seeds, either individual pixels or regions, and give ways in which this can be done manually or using automatic procedures.

Griffin et al[61] define an agglomeration algorithm which, starting with single pixel regions, merges pairs of neighbouring regions using a homogeneity criterion.

Region growing techniques are a powerful method which have been applied in numerous ways to the segmentation problem and new variations are continually being developed.

- ***Split and Merge***

Split and merge methods seem to lie at the opposite extreme to region growing as they start with the whole image as an initial segment. As originally defined each current segment is split into quarters if its homogeneity does not meet some defined criterion. Each new quarter, if itself homogeneous, is either merged with any neighbouring region which will produce a homogeneous larger region, or left unaltered. Later versions consider the splitting of arbitrarily shaped regions. Generally a graph tree structure maintains the relationship between regions at each stage of processing. Some of the techniques used to merge regions are equivalent to those used in region growing.

Browning and Tanimoto[75] describe a split and merge scheme operating, in the first place, on non-overlapping subimages and then combining the results to eliminate any artificial boundaries. Gevers and Kajaovski[60] use a system based on Delaunay triangulation to direct region division.

Split and merge methods have difficulty dealing with small subregions within large uniform areas as these can easily be missed if the test for inhomogeneity is not sufficiently sensitive.

- ***Classification***

Classification as used in segmentation is the assignment of a particular label (class) to each pixel. The label assigned is dependent on the information measured about each pixel which may be as simple as the original grey level or a more complex 'feature vector' calculated using the grey level values in a surrounding local window of pixels. Considerable use of Bayes theorem is often made (see later) where classification is dependent on knowledge of the class-conditional probability density function or pdf (the probability of a particular feature vector being measured for each known class). These pdf's might have a known functional form (often a Gaussian is assumed) with only some parameters unknown and having to be estimated - a 'parametric' classifier. If the functional form of the pdf's is unknown then the classifier is 'non-parametric' and the form must be estimated from a considerably larger set of training data than required for a parametric type. This process of estimating the class-conditional pdf's is termed *training* the classifier.

'Supervised' training requires that the different classes are defined by the user and their pdf's are known or measured prior to application. This has been used to partition land images into groups such as trees, buildings, water, etc; using cluster-centre vectors derived by training from other manually segmented images. This is scene segmentation (see Section 1.1) but the classification can be in terms of 'dark' and 'bright' areas or other abstract qualities. Tou and Gonzalez[76] define many of the statistical techniques available for pattern classification.

Kumar and Desai[30] use Bayesian statistics (see later) and compute the conditional probabilities for a small number of image feature types; 'road', 'grassland', etc; using a known image to then 'interpret' (= segment) other images of the same type. Kim and Yang[33] describe a scheme minimising an energy function to produce an optimum classification.

Bouman and Shapiro[45] classify regions of distinct statistical behaviour which are characterised using a Markov Random Field structure.

Classification techniques in this strict sense are obviously restricted to segmentation into a rather narrow range of user-defined classes which have to be common to all the images being used and might well be totally inappropriate to another type of imagery.

- ***Clustering***

'Unsupervised' training estimates the class-conditional pdf's without requiring the user to define the number of classes or their characteristics. These are determined by locating clusters of points in feature space - 'cluster analysis'. Clustering places pixels into self-similar groups without necessarily placing any meaning on them whereas supervised training classification techniques (often simply termed 'classification') use user-defined cluster centre vectors to restrict the pixels to given classes corresponding to a chosen set of user required region types. (However, it is evident from the studied literature that this distinction between clustering and classification is not one adhered to by all authors and often the terms are used interchangeably.)

A variety of statistical methods are available for grouping the feature vectors, and hence their associated pixels, into the most compact possible sets. An important parameter here is the allowed number of clusters which can be adjusted within the algorithm. As all the pixels in one cluster are not necessarily spatially connected, the number of resulting regions will usually be larger than the number of clusters, sometimes considerably so. The work of Khotanzad and Bouarfa[12] is based on analysing the multi-dimensional histogram of the feature vectors rather than the vectors themselves and is totally automatic.

The feature vector used in clustering is often complex and much work has been undertaken to segment textured images with vectors representing the texture. Hofmann et al[19] describe a method for segmenting textured images which is unsupervised. Textures are characterised using Gabor filters (see later) and clustered using functions which are optimised using simulated annealing. Bouman and Liu[25] also segment textured images using an unsupervised parameter estimation algorithm to determine the number of different clusters and their associated parameters.

Ghosal and Mehrotra[46] calculate Zernike moment based feature vectors for their range imagery and then cluster this information to segment the images into different surface types.

Beaulieu and Goldberg[48] postulate a hierarchical stepwise optimisation scheme which sequentially reduces the number of clusters by merging until 'the required number' is obtained. Li et al[65] use simple user-defined grey level clustering in an otherwise rather complex mathematical procedure.

Clusters are easier to recognise when they contain many similar points but minor regions or uncommon objects may be overlooked and this is therefore an inherent failing of this particular method.

- ***Neural networks***

Neural networks are arrays of simple processing elements which are connected in a layered network architecture. The interconnection weights of the processing elements are adjusted to an optimum by training against a known set of feature vectors and the neural network can then be used to classify pixels in new images.

Bhattacharya et al[22] formulate a texture segmentation scheme using a multi-layer perceptron - a specific type of neural network - that works without the need to select a feature set. Kohonen et al[43] use a self-organising map in a number of applications including the classification of textured images and Raghu et al[39] also perform unsupervised texture classification with a combination of deterministic relaxation and neural networks.

- ***Snakes***

Snakes are deformable contours enclosing a gradually changing group of pixels. Their deformations are governed by a function which is usually in two parts; the first determines the allowable contortions of the contour and the second controls the homogeneity of the enclosed pixels.

Chion and Hwang[57] use a neural network guided snake to delineate distinct features in MRI brain images. Defining the snakes energy function in terms of 'tension', 'stiffness', 'pressure' and 'potential' Ivins and Porrill[63] produce a region growing technique known as an 'Active Region Model' or 'Statistical Snake'.

Snakes are particularly useful for tracking moving boundaries in an image sequence where the changes of the snake contour are small from one image to the next, but they

can be used for the more general segmentation problem of extracting multiple regions in a single image.

- ***Genetic algorithms***

Genetic algorithms are search methods whose functioning is inspired by the genetic laws governing natural selection and evolution. They are especially good at avoiding local minima and this has made them a powerful alternative to more conventional optimisation algorithms. Genetic algorithms are not by themselves segmentation methods but are frequently combined with other techniques to produce more powerful complete systems.

Andrey and Taroux[38] implemented a segmentation method using classification in which the labelling is iteratively modified using a genetic algorithm. Bhanu et al[8] describe an adaptive segmentation system which incorporates a feedback loop consisting of a machine learning subsystem, a segmentation algorithm and an evaluation component. The learning component, based on a genetic algorithm, improves the segmentation performance.

Genetic algorithms are multi-point derivative free optimisation techniques which are generally robust but often very slow.

- ***Texture***

Texture has proved very difficult to define precisely but is concerned with the spatial distribution of grey level variations. A description of some texture properties has already been made (see section 1.2). Examples of rigid, fully determined textures include a simple binary chessboard pattern and the sequence of bricks and mortar in a wall. Less rigid textures exist in pictures of leaves on a tree or the grain in a piece of wood. The common factor of all scenes with texture is the existence of a spatial variation in tone which consists of the distribution of similar elements over a background. The size of the texture elements may vary according to some probabilistic distribution and there may be more than one type of element each with its own distribution. Furthermore the grey levels of the pixels within each element type may follow another distribution. When there is only one type of texture element whose size

is that of a single pixel and whose grey levels follow a given probability distribution this is the simplest texture: noise.

Many authors have been concerned with establishing satisfactory measures of texture with the aims either of classifying different parts of an image or of achieving segmentation. Reed and Du Buf[78] in reviewing texture feature extraction methods group them into three classes: *feature based* in which some characteristic of the texture is chosen and measured; *model based*, which hypothesise some underlying process of production of the texture and attempt to measure its underlying parameters; and *structural,* which assume detectable primitive elements which are arranged according to placement rules. Some textural features which have been used are described below:

First order statistics, such as variance, skewness, and others, such as contrast and entropy which are derivable from the histogram, are useful in distinguishing the simple noise textures. However these, by definition, have no spatial variation and might even not be regarded as real textures. It is very easy to define different true textures which have very unlike spatial relations but the *same* first order statistics and which are therefore not separable by these measures.

The *Laws Metrics* consist of a set of small (e.g; 5 by 5) *microtexture* filter masks which are convolved with the image. Statistics for each pixel in the filtered image are then computed in a surrounding neighbourhood and used as texture features. Hsiao and Sawchuk[79] used a selection of the Laws metrics outputs in their segmentation method.

Fourier transforms also contain texture information. A strong Fourier component is an indication of the presence of a texture with grey levels repeated at a spatial distance and direction given by the component's position. Reducing a Fourier transform to a more manageable set of features can be done either by locating peaks in the transform or averaging the transform over ranges of polar coordinates to give a set of numbers corresponding to different texture frequencies and directions. Haralick[80] reviews texture methods using orthogonal transforms including the Fourier transform and he concludes that their performance is generally poorer than features based on second-order statistics as calculated by, for example, the grey-level co-occurrence matrix (see below).

The *fractal dimension* of an image region has been found to give a good indication of textural properties. An efficient method of expressing the fractal dimension is by calculating the Hurst coefficient. Windows are moved about an allowed neighbourhood of each pixel and the range of intensity values measured in each. The average range of image intensities is found and this is plotted for different window sizes. The gradient of the plot is the Hurst coefficient. Russ[81] gives examples of the Hurst coefficient used to segment electron micrographs of liver cells and pictures of fractured steel surfaces.

The *Grey Level Co-occurrence Matrix, GLCM,* is an extension of the first order histogram and is based on the number of pairs of pixels set at defined grey levels. A pixel separation vector, $\underline{d} = (d_x,d_y)$, is established, defined in both direction and magnitude (number of pixel units horizontally and vertically) in an image. The i,jth element of the co-occurrence matrix, $S(i,j,\underline{d})$, is the number of times that a pixel pair occurred in which the first pixel had grey level i and the second j. A GLCM can thus be calculated for each possible separation vector and has dimensions G*G where G is the number of grey levels used (this is sometimes reduced from the original number in the image, possibly after a process of 'histogram equalisation'). Haralick et al. (1973)[82] propose and evaluate a number of measures calculated from the GLCMs.

Other methods which are frequently used in texture segmentation are Wavelets, Markov Random Field Models and Bayesian Estimation. However these techniques are all much broader in scope and hence are given separate sections below:

- ***Wavelet transforms***

Wavelets have provoked considerable interest in the past few years. Unlike the Fourier transform which is based on sinusoidal waveforms extending to infinity in both directions, the wavelet transform utilises basis functions having a limited extent (hence the name). This is in many ways more satisfactory for analysing real signals and images which themselves have only finite size. A complication of the wavelet is that, besides defining the equivalents of frequency and amplitude, it is also necessary to fix the position ie; where the centre point of the wavelet is located.

Texture segmentation with wavelets is often achieved using a bank of filters whose composite output constitutes a feature vector which can then be used to classify the texture. Dunn and Higgins[26] concentrate on the design of the optimal Gabor filter (a

related concept) for the discrimination of just two textures. Laine and Fan[27] use wavelet signatures corresponding to different scales and orientations as the input to a neural network classifier to achieve virtually 100% correct classification of 25 different textures. Pichler et al[21] use multi-channel Gabor filtering and reduce the feature space by a criterion of feature contrast. Porter and Canagarajah[28] combine optimal feature selection from wavelet analysis with clustering. Raghu and Yegnanarayana[20] and Hofmann et al[19] both use Gabor filters and Markov models to cluster their extracted feature sets into a final segmentation.

- ***Bayesian Estimation***

Bayesian estimation applied to images is concerned with the assignment of classes (labels) to each pixel given a 'feature vector' of information about the pixel. Bayes's Theorem defines the 'a-posteriori' probability; the probability of being in a defined class given a certain measured feature vector, in terms of the 'a-priori' probability of being in a given class and the class-conditional probability; the likelihood that, for a defined pixel class, a given feature vector will be measured. Typically one wishes to choose the 'best' set of labels by maximising the overall 'a-posteriori' probabilities.

In perhaps the simplest application the classes are taken to be 'background' and 'objects' and the feature vector as just the grey levels themselves. The a-priori probabilities might be taken to be fixed at the known (or assumed) fractions of background and objects present in the image and the class-conditional probabilities assumed to be Gaussian distributions with a known mean and variance for each class. With these assumptions the a-posteriori probabilities are maximised for any grey level by simply choosing the class for which the product of a-priori and conditional probabilities is greatest.

Complications to this basic model arise in two ways; firstly there may be more than two classes. Indeed the number might be unknown and as a consequence the class-conditional probabilities are also not known fixed functions. Part of the task might be to choose the optimum number necessitating a use of 'unsupervised' statistical clustering methods (see earlier).

The second complication occurs when the a-priori probabilities are sensibly taken to have a dependence on the classes already assigned to neighbouring pixels. It might, for

example, be known that the 'objects' in the above two-class image tend to have a minimum size of (say) 100 pixels. In this situation a single pixel labelled (on the strength of it's grey level alone) as 'object' with surrounding 'background' neighbours would seem to be mis-classified. A method of dealing with this is described in the next section

- ***Markov Random Field***s

The Markov Random Field is seen by Dubes et al[32] as 'a convenient means for introducing context, or dependence, among neighbouring pixels'. Furthermore; 'context is important in image segmentation because contiguous pixels are likely to belong to the same region'. Essentially a matrix of probabilities is imposed for determining the class of the current pixel being investigated dependent on the classes of its neighbours.

Geman and Geman[34] were amongst the first to formulate a precise mathematical description of the application of Markov random fields to the problem of the Bayesian restoration of images corrupted by noise. Their work also 'exploits the equivalence between Markov random fields and the Gibbs distribution'. The Gibbs distribution is an exponential expression of the joint probability function of the pixel labels. The (negative) exponent is an 'energy function' dependent in part on the parameters of the class-conditional functions but also in part on the spatial interaction of the classes of neighbouring pixels. The spatial interaction model is chosen by the user and can be as simple as an on/off switch depending on whether a neighbouring pixel has been assigned the same or a different class. Minimising the energy function is equivalent to maximising the joint a-posteriori probabilities and this is done by a correct choice of class for all the pixels.

As Besag[35] states 'scenes typically contain perhaps 256 by 256 pixels so the computational burden of the optimisation is enormous. Dubes et al[32] compare three methods of accomplishing this optimisation problem; iterated conditional modes, maximiser of posterior marginals and simulated annealing, concluding that the first was the most robust and performed as well as the others. However their application was somewhat limited involving mainly only noise corrupted images for which the noise parameters and the number of classes had to be user-supplied.

Derin and Elliott[31] use a 'hierarchical Gibbsian model' of two Markov processes. The higher level processes - region formation - are the same for both noisy and textured images but the low-level processes are different. LaValle and Hutchinson[42] apply Markov random fields to the problem of determining a criterion for region merging where the region models are parametrised but the parameter values do not need to be estimated. Snyder et al[69] segment magnetic resonance images by first using Markov random fields to form functions which 'resemble the data', 'are locally homogeneous' and 'only have prespecified brightness values' and then optimise the combined function using mean-field annealing.

- ***Scale***

The idea that information in images occurs at different scales has been promoted by various authors for a considerable period. Marr and Hildreth[55] were concerned with constructing the 'primal sketch' - the hypothetical results of early human visual processing stages. They formulated a method that detects intensity changes at different scales. The different scaling is produced by convolving the image with a series of Gaussian shaped filters of different widths (standard deviations). The intensity changes are detected by the presence of a zero crossing in the second derivative.

This technique of Gaussian blurring has been widely used but, as Griffin et al.[10] point out, whilst an attractive method it 'has an inherent difficulty in that it forms a continuous hierarchy of different scales which require further processing to discover the significant levels'. They describe a novel method designed to produce only meaningful scales which is most easily envisaged in one dimension. Maxima and minima are first found and used to split the image function into districts. Joining the centroids of each district produces a new function which has automatically eliminated the lower order peaks and troughs to produce the new scale. Implementation in two dimensions is considerably complicated by the need to detect saddle points and the use of maximum gradient paths to join the points and of Delaunay triangulation to fit the new points at each scale with a new function.

Texture in images, as has already been noted, has an inherent scale and therefore detection of texture presupposes either that the correct scale is known or that a multiplicity of scales must be used and analysed to find those that are significant.

Bouman and Liu[25] use Markov random fields as a model of texture and proceed by segmenting images at different scales proceeding from coarse to finer resolutions until finally the individual pixels are classified. This work is pursued in Bouman and Shapiro[45] in which the Markov random field is replaced with a novel multiscale random field.

Tabb and Ahuja[11] have a totally new approach by arguing that 'the issues of scale selection and structure detection cannot be treated separately'. They define a new transform that represents image *structure* at different scales and not the image itself. Their novel non-linear transform also only 'sees' the structure present at the current scale (and not at all the higher scales as well). The algorithm is entirely unsupervised and some visually impressive results are presented.

- ***Summary***

The above represent many of the algorithms available to date for the purpose of image segmentation. Some of these use, directly or indirectly, a-priori information of the properties of the objects which it is desired to find - the 'scene segmentation' which was described in Chapter 1. Others are concerned with the direct application of textural finding methods. Such ideas are not of direct relevance here where the primary concern is to segment into regions which have only a smooth variation of intensity and to do this entirely without the use of contextual information. It is the intention to produce a method that directly connects neighbouring pixels that are part of the same smooth surface. Some of the above techniques do utilise grey level information and local statistics to place pixels in various different groupings - the thresholding, region growing and classifying methods for example - but the relationship with selecting on the basis of smooth surface characteristics is at best only indirect and often inexact. Methods which directly use a surface fitting technique are of strong interest here and these are reviewed in the next section.

3.2 Segmentation by surface fitting

Fitting an intensity function to a region of pixels is a technique known as 'surface fitting'. This type of technique is of particular interest to this thesis which is partly based on fitting within a local area.

One of the first to use such a technique for segmentation was Pong[17] who with others, including Haralick, used a 'facet' model to fit grey level data. This work was in fact a continuation of that started by Haralick and Watson[49]. The facet model, as they describe it, assumes that the image domain is composed of connected regions called facets in each of which the grey levels are a polynomial function of the pixel position. Allowing for a noise value of n(r,c) at pixel position (r,c) the grey level g(r,c) is taken to be well represented by:

$$g(r,c) = \sum_{i=0}^{N} \sum_{j=0}^{i} a_{ij} r^j c^{i-j} + n(r,c)$$

where the a_{ij} are constant coefficients of the row (r) and column (c) pixel position powers. N is the degree of the polynomial used; N=0 is the 'flat facet model' and N=1 the sloped plane model. In any given window of pixels the least squares method can be used to minimise the error of fitting.

The window with the best fit is chosen and the pixel grey level replaced with that given by the fitted polynomial. Reiterating this procedure results eventually in all pixels being tied to a particular polynomial surface and was proved to converge. A statistical method is then used to determine the significance of differences between adjacent polynomial surfaces. Thresholding this statistic effectively produces edges and hence a segmentation. Only the flat and sloped (constant gradient) models were tried. Following this initial segmentation a region growing procedure was applied, merging regions providing a defined list of grey level statistics as recorded in 'region property vectors' were sufficiently similar.

In later work, Haralick[9] uses the same facet model idea again to produce edges, this time defining the edge to occur within a pixel if and only if there is some point in the pixel which has a negatively sloped zero crossing of the second directional derivative in the direction of a nonzero gradient at the pixel's centre.

Besl and Jain[18] formulated a variable-order surface fitting method. Their purpose, in part, was to avoid the restrictions implicit in many other segmentation techniques especially the reliance on assumptions about the underlying structure of the sensed image data. They quote Witkin and Tenenbaum[77] who argue that *perceptual*

organisation mechanisms exist in the early stages of human visual processing that are independent of high level knowledge for correct image interpretation.

Their method involves two stages. The first is the creation of a surface label type for each pixel. The mean and Gaussian curvature values are computed and their signs alone (-, 0, +) used to define a set of eight possible surface types: peak, pit, ridge, valley, saddle ridge, saddle valley, flat, minimal. Grouping together connected pixels with the same surface type produces an initial 'coarse' segmentation which is then modified by the second stage.

The second stage of the method is an algorithm of some considerable complexity but the essentials are as follows. A 'seed' region is chosen by shrinking the next largest region using a binary image erosion operator. A seed region is initially fitted to a planar polynomial (i.e; of degree one) and, if the fit passes a test of accuracy, the region is grown by adding pixels providing that they are 'compatible' which means that their values and derivatives are 'close' to those predicted by the polynomial. After accumulating as many pixels as possible in this way the new enlarged region is refitted to a polynomial of the same order. If this fit is acceptable then, using the new polynomial coefficients, region growing is attempted again. This process is repeated until the region ceases to grow. If a newly grown region fails to fit to the same order of polynomial sufficiently well, the order of the polynomial is increased (e.g; planes become quadratics) up to the level of a quartic - hence the 'variable-order' of the title. As the algorithm proceeds, pixels can be relabelled as belonging to a new region.

The end result of the process is a partitioning of the image into regions of arbitrary shape (= a segmentation) each of which is approximated by a low-order bivariate function. The technique was originally intended for use with range images in which the grey level is a measure of the distance from the sensor of the part of the surface represented by the pixel but Jain and Besl stress that it is capable of segmenting any type of image that can adequately be represented as a noisy sampled version of a piecewise-smooth graph surface and they clearly consider ordinary visually sensed images to be of this type.

Trucco and Fisher[16] also use the mean and Gaussian curvatures to qualitatively segment range image data. Their work includes a Gaussian smoothing procedure with a

varying kernel based on a 'diffusion' paradigm. Considerable effort is made to assess the accuracy of the classification achieved using synthetic images and the effects of quantisation and smoothing.

Also working with Jain, Liou and Chiu[37] developed a scheme for 'Signal-Level Perceptual Organisation' whose objective is 'to generate a collection of abstract representations (in terms of region models) from the signals at the pixel level such that the content of the image is maintained'. This description is then envisaged as providing a basis for later scene interpretation.

Their technique consists of three stages and emphasis is laid on the parallel nature of the methods of the first two stages. The first stage consists of obtaining an edge image using a simple Sobel edge detector which is then multi-thresholded at thresholds α_i - a process termed 'α-partitioning'. A grey-level α_i-image is produced by thresholding at an edge strength α_i and then grouping connected pixels to form a set of 'region hypotheses'.

The second stage consists of 'region filtering' in which each α_i-image is first filtered to 'verify the correctness of the hypothesised models'; a process which is done by fitting the original pixel grey level values of each region in turn to a quartic bivariate polynomial. If the fit is not satisfactory (a sophisticated statistical test is used) the region is 'filtered out' of this partition.

The final stage consists of combining the filtered α_i - partitions which is achieved by simply forming an inclusive image 'OR' on the whole set. This is shown to produce a set of regions which satisfies both the uniformity predicate (in this case meaning that regions are 'good fits' to the chosen polynomial and combinations of neighbouring regions are not) and the condition that the set be complete such that it covers the whole of the original image.

3.3 Summary

This thesis concurs with the premise of Besl and Jain[18] in which they state: *it should be possible to group pixels in many types of images using only low-level information* and

further that; *perception of surfaces is a low-level grouping operation that plays a fundamental role in many image understanding tasks.*

The surface fitting methods described above, whilst partly fulfilling this premise, are unsatisfactory for various reasons: Pong[17] working with Haralick produced a superfluity of regions by overthresholding their edge detector and then merged these using a somewhat arbitrary region property vector, effectively losing the surface fitting information and starting afresh with a different set. Other work by Haralick[9] was primarily concerned with producing edge detection algorithms and only secondarily with achieving segmentations.

The work of both Besl[18] and Liou[37] was concerned with producing not just a segmentation but, in addition, a description of each region in terms of the polynomial coefficients used in the eventual fitting. Whilst undoubtedly a useful additional piece of information for some succeeding work, this does seem unfortunately restrictive. In order to satisfy the requirement of fitting an intensity function to each whole region a sufficiently complex function needs to be chosen (e.g; a quartic). This complexity means that there is the possibility that smaller areas covering a true region boundary might fit the function and be output as a single region. In addition the process of fitting and refitting these complex functions to changing groups of pixels is a markedly complicated task.

It is the aim here to cause a grouping of pixels by estimating the degree of connectivity of all neighbouring pixels using the similarity of their best fit surfaces as measured in local areas surrounding each pixel. The matrix of connectivities so achieved can then be used to define an ideal segmentation. Ways of approaching this ideal and of dealing with the attendant problems are specified in the following chapters.

4. Segmentation using Pixel Linking

4.1 Overview

The fundamental concept of the present work is that all images can be segmented into regions each of which has the characteristic of a smoothly varying grey level. Although the variation is smooth, the total change over the whole region might well be quite large and too complex to be represented by any simple function. For this reason it is not attempted to accumulate whole regions directly. Instead, the likelihood of neighbouring pixels being part of the same or of different surfaces is first established. These likelihoods are termed 'beliefs' as they are based on empirical functions rather than derived as exact probabilities. The beliefs are calculated by finding the characteristics of the local surfaces for each pixel and then comparing neighbouring surfaces and deciding on their similarity. The term belief has been used by other authors and in particular by Dempster and Schafer in their work as described by Schafer[83] and later the similarities between their definitions and those made here will be drawn.

These local surfaces are found by fitting a simple mathematical function to all small groups of pixels and, for each pixel, choosing that surface which produces the least error of fitting. For the purposes of this work square windows of pixels have been chosen to use for the local surface fitting although there is no real restriction here and indeed, in future, more sophisticated work, it might be possible to have a scheme which varies the shape and size of the local group according to the situation.

The size of window used is restricted in various ways: firstly, too small a window would not allow a sufficiently accurate measurement of the surface parameters; for example a two by two window would be greatly affected by any imperfections in the surface. Secondly, the size of window needs to be sufficiently large to allow the chosen function to be adequately fitted; again a two by two window would allow the local surface gradient coefficient to be estimated but would not have sufficient information to enable higher order coefficients to be estimated. A third restriction is that too large a window size would mean that some pixels on the inside of the convex borders of a

region would not lie within any window which is totally covered by the containing region and hence an accurate representative local surface could not be calculated.

The function used for the fitting is also sensibly restricted; it needs to be complex enough to allow a sensible comparison of neighbouring surfaces but not so complicated that virtually all pixel collections would be accurately fitted whatever their values.

A computational limitation is also imposed by the fact that the larger the window and the more complex the function used, the longer the computing time required.

For the purposes of this work a combination of a biquadratic function with a 5 by 5 window has been chosen as this was felt to satisfy the above restrictions and was a computationally practical solution as well.

The best fitting surface is chosen for each pixel as represented by its coefficients. A belief of fitting is calculated based on the error of the fitting. All neighbouring pairs of pixels are then examined and a belief of their being linked, that is, that they are part of the same surface, is calculated based on the fitting beliefs of the two pixels and on the additional belief of the two surfaces being in fact the same. The details of this process follow in the succeeding sections.

4.2 Pixel Fitting Beliefs

As already stated the function used in this work for the purpose of representing the local grey level surface is a biquadratic surface which has the form;

$$f_i = c_{00} + c_{10}x_i + c_{01}y_i + c_{11}x_iy_i + c_{20}x_i^2 + c_{02}y_i^2 \qquad \text{Eq 4-10}$$

where the f_i is the fitted pixel grey level to the ith pixel at position (x_i, y_i) and the c_{ab} are the coefficients that are to be determined from the fitting.

This can be rewritten in the form of the matrix equation;

$$\mathbf{F} = \mathbf{BC}$$

where, in terms of N, the number of pixels in the window;

$$\mathbf{F} = \begin{bmatrix} f_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_N \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} c_{00} \\ c_{10} \\ c_{01} \\ \vdots \\ c_{02} \end{bmatrix},$$

and

$$\mathbf{B} = \begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 & x_1^2 & y_1^2 \\ 1 & x_2 & y_2 & x_2y_2 & x_2^2 & y_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & y_N & x_Ny_N & x_N^2 & y_N^2 \end{bmatrix}$$

which are respectively, a (N*1) column vector of the N fitted pixel grey level values, a (6*1) column vector of the coefficients and a (N*6) matrix dependent on the pixel coordinate position values.

If g_i are the actual pixel grey level values then the mean square error (MSE) of the fitting is;

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left[g_i - f_i\right]^2$$

Or, defining the (N*1) column vector of actual pixel values,

$$\mathbf{G} = \begin{bmatrix} g_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ g_N \end{bmatrix}$$

$$\text{MSE} = \frac{1}{N}\mathbf{E}^{\mathrm{T}}\mathbf{E}$$

in matrix form, where,

$$\mathbf{E} = \mathbf{G} - \mathbf{F} = \mathbf{G} - \mathbf{BC}$$

is the (N*1) column vector of fitting errors and $\mathbf{E}^{\mathrm{T}}$ is its transpose .

A least squares fitting is achieved by differentiating with respect to the elements of **C** and setting the derivatives to zero. This leads to the solution,

$$\mathbf{C} = \left[\mathbf{B}^{\mathrm{T}}\mathbf{B}\right]^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{G}$$

as the vector of coefficients which minimise the mean square error. This can be rewritten as;

$$\mathbf{C} = \mathbf{PG}$$

in terms of the square matrix, **P** ,the *pseudoinverse* of **B** ;

$$\mathbf{P} = \left[\mathbf{B}^{\mathrm{T}}\mathbf{B}\right]^{-1}\mathbf{B}^{\mathrm{T}}$$

For this work a 5*5 window was chosen for the fitting and the coordinates expressed relative to the centre of the window; x_i ,y_i = -2,-1,0,1,2. For convenience, taking the pixels in column order, starting at position (-2,-2), the matrix B, now 25*6 in size, becomes;

$$\mathbf{B} = \begin{bmatrix} 1 & -2 & -2 & 4 & 4 & 4 \\ 1 & -2 & -1 & 2 & 4 & 1 \\ 1 & -2 & 0 & 0 & 4 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 2 & 2 & 4 & 4 & 4 \end{bmatrix}$$

The biquadratic function, Eq 4-10, is fitted to the twenty five pixels in each 5*5 window in the image and the coefficients, individual pixel errors and root mean square

error calculated using the methods outlined above. The windows are moved one pixel at a time over the whole image and the coefficients and root mean square errors, Δ_{rms}, stored in a set of output images at the position of the window centre pixel.

Each pixel in the body of the image is a member of 25 such windows (with a lesser number for the outer two rows and columns of pixels).Windows which are wholly contained within a single surface (region) will produce fittings with root mean square errors that are indicative either of the noise present in the image or of the suitability of the biquadratic function as a representation of the particular surface present. An analysis of the problems arising from these considerations will be given in more detail in Chapter 5.

A window which straddles a boundary of two surfaces will usually produce errors that are relatively higher and it is this fact that enables the 'best' window to be chosen for each pixel. This best window for pixel, p, is taken to be that with the lowest mean square error as this represents the best fitting surface, S_p. The coordinates of the pixel relative to this window centre, x_p,y_p, are stored at the pixel position in the output image.

All of the different fitting coefficients, c_{ij}, and their associated root mean square errors Δ_{rms}, are stored at the coordinates of the window centre pixel. However only the individual pixel error, ε_p, in the 'best' window, is stored along with its coordinates in this window; x_p, y_p

The information stored at this stage of the program can be represented diagrammatically as in Figure 4-1.

It should be noted that although the coefficients and root mean square errors can only be calculated for positions at least three pixels from the image edge nonetheless all pixels have an associated 'best' surface to which they have been assigned.

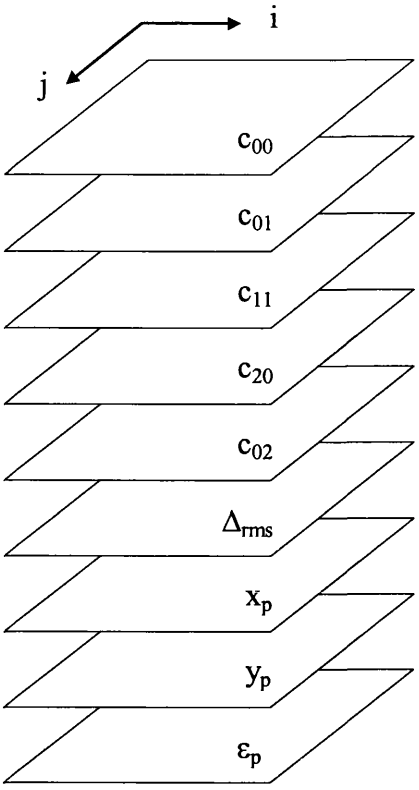


Figure 4-1 Information stored
by fitting process

A test image, Figure 4-2, consisting of a variety of exact quadratic surfaces, was produced in order to verify the program output. The image is a strip, presented horizontally, of 128 by 16 pixels divided into eight sections each of size 16 by 16 pixels. These sections comprise the following with their defining equations for the pixel grey level values, g(x,y) given relative to an origin at the top left corner of each section, x=0,y=0:

1. pixels 0 to 15, a constant area; g(x,y)=127

2. pixels 16 to 31, a ramp rising from 127 to 255; g(x,y)=127 + 8x

3. pixels 31 to 47, a constant area; g(x,y)=255

4. pixels 48 to 63, a ramp decreasing from 255 to 15; g(x,y)=255 -15x

5. pixels 64 to 79, a constant area; g(x,y)=15

6. pixels 80 to 95, a quadratic surface in x rising from 15 to 240; g(x,y)=15 + x^2

7. pixels 96 to 111, a mixed product surface rising from 0 in the top left corner to 225 in the bottom right; $g(x,y)=xy$

8. pixels 112 to 127, a quadratic in x decreasing from 225 to 0; $g(x,y)=225 - x^2$

The remaining parts of Figure 4-2 together with Figure 4-3 show the pixel values taken from a line sampled across the strip along the fourth row down. Points to note include:

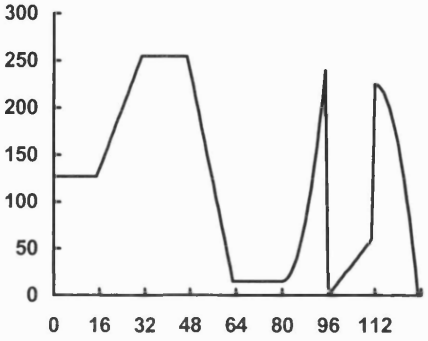
- Firstly the calculated coefficients do not always equal the generating coefficients as might at first have been expected. This is because the generated functions are made with respect to a single coordinate system with an origin situated at the top left corner of each image section whereas the origin for the calculated coefficients is situated at and moves with the centre of the 5 by 5 window used. As a consequence a calculated coefficient is the same as the generating coefficient only when there is no higher order coefficient present in the generating function. For example within the ramp sections the first order x-coefficient equals the generating coefficient but the constant coefficient, zero in the generator, gradually increases in the fitted version.

- c00, the constant value term, approximately follows the form of the original pixel values within the sections, with more obvious deviations at the junctions where the function fitting is not perfect.

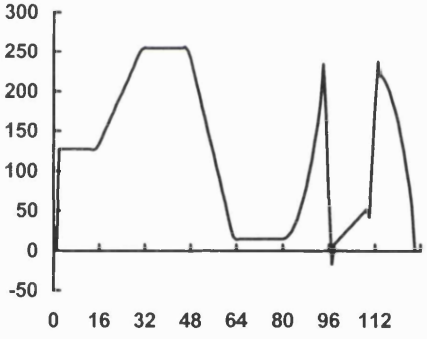
- c10, the gradient in x, is constant within the two ramp sections (2 and 4) with values equal to their gradients (+8 and -15). In the quadratic sections (6 and 8) it increases linearly.

- c01, the gradient in y, is zero everywhere apart from the seventh section which is the only one with any change in the y-direction.

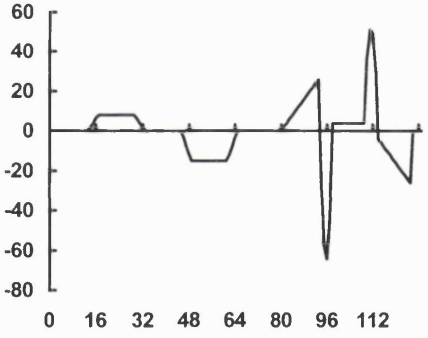
original pixel values



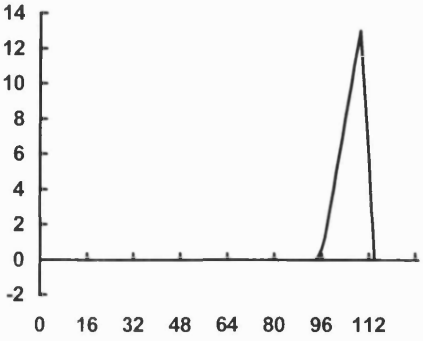
c00



c10



c01



c11

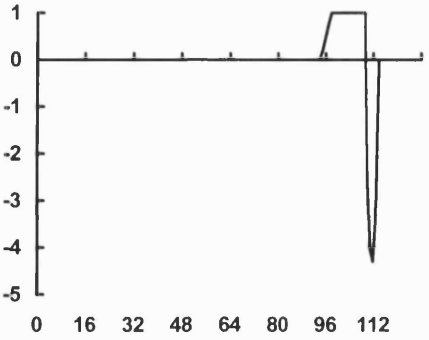


Figure 4-2 Test image and results

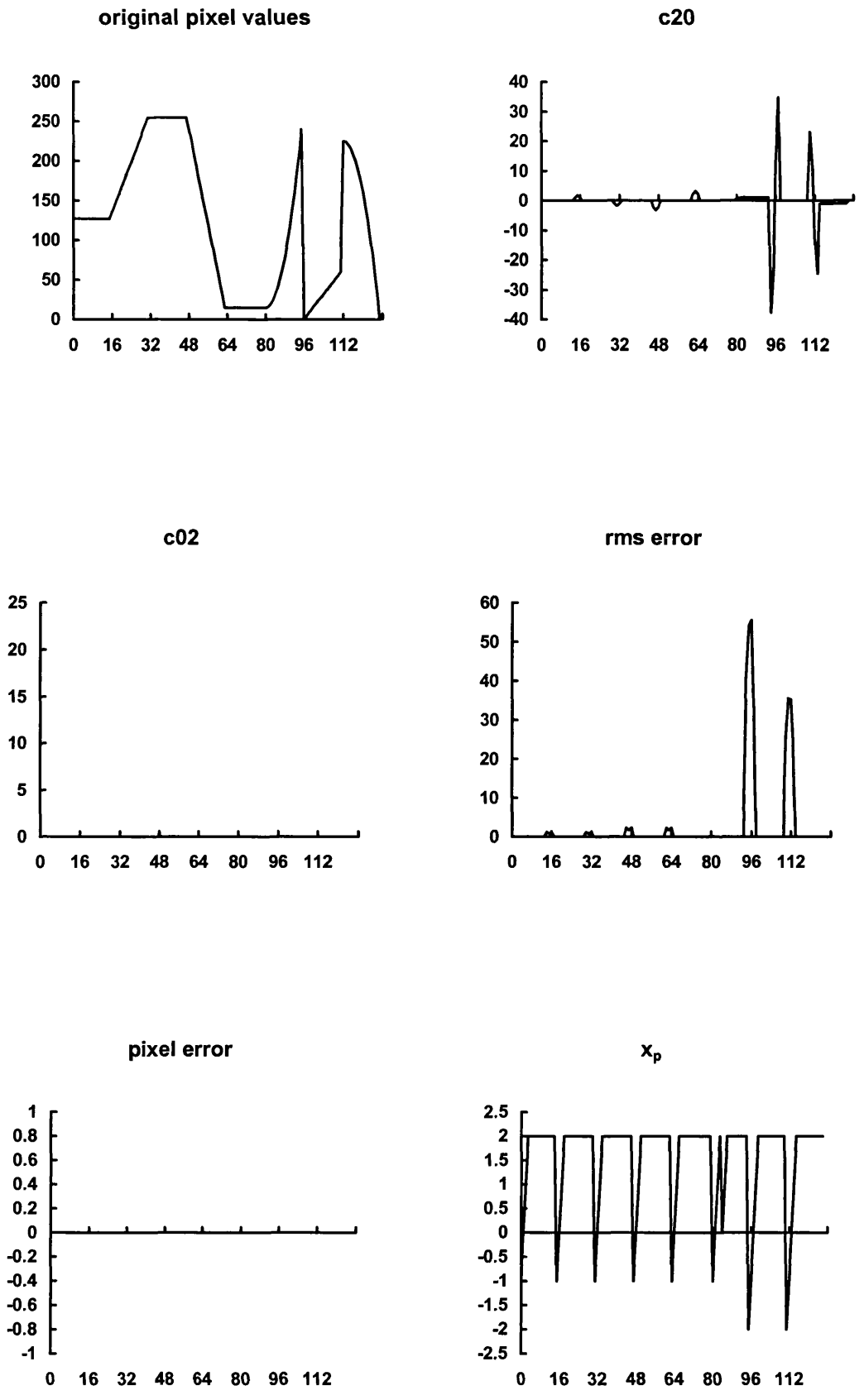


Figure 4-3 Further test image results

- c11, the x-y product coefficient is also zero everywhere apart from the seventh section where it takes the value of the generator - one.

- c20, the quadratic coefficient in x, takes the generating values of +1 and -1 in the quadratic sections (6 and 8) and is zero elsewhere within the other sections but has spikes at the junctions of sections.

- c02, the quadratic coefficient in y, is zero everywhere which is to be expected as there is no term of this order in the image.

- Δ_{rms}, the root mean square error term is zero within all sections because an exact fit can be made as the functions are perfect quadratics. At the junctions various values are taken.

- ε_p, the pixel error term, is zero everywhere including at the section junctions, as each pixel is always a member of (at least one) perfect fitting surface.

- x_p, the pixel x-coordinate in the best fitting window. This jumps from a value of +2 at the right-hand edge of each section to a value of -1 or -2 at the start of the next section showing that the best fitting windows do not overlap the section junctions. Within the sections all the windows are perfect fits so any value could in theory be taken but two is favoured simply because it is found first and is not updated as the error does not improve.

The belief of fitting, $B(p \in S_p)$, of the 'best' surface, S_p, as given by the coefficients is now defined. An empirical function is needed with the following properties: (The root mean square error (rms) of this surface, $\Delta_{rms}(S_p)$, is denoted as Δ_p for convenience.)

- Values which range between 0 and 1

- A proportionally lower value for higher rms error of fitting values, Δ_p.

- A value of 0.5 for an error, Δ_p, equal to a user-chosen 'noise' level, σ.

The function chosen was:

$$B(p \in S_p) = \exp\left\{-\ln(2)\left(\frac{\Delta_p}{\sigma}\right)^2\right\} \qquad \text{Eq 4-11}$$

which gives the belief, $B(p \in S_p)$, for pixel p. This function has the above required properties and is reasonably uncomplicated as well as being in a form which lends itself to some analysis and is plotted in Figure 4-4.

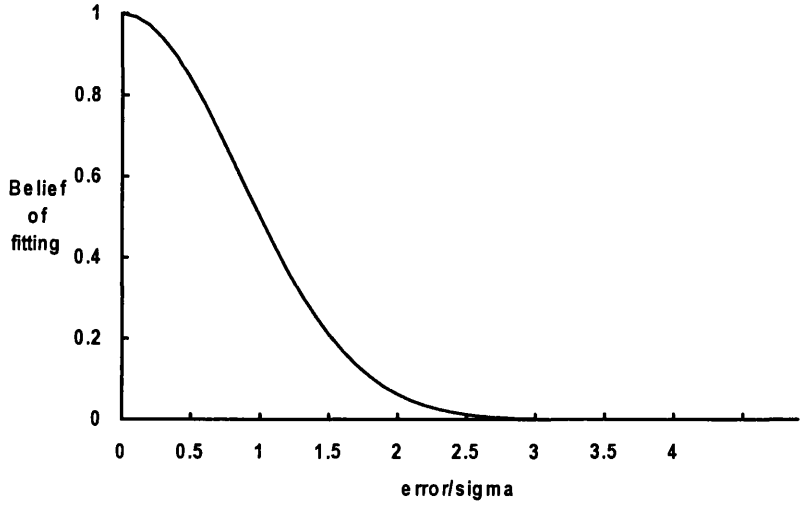


Figure 4-4 Belief of fitting

It is important to be clear about the meaning of the function $B(p \in S_p)$. It is the belief that the pixel, p, is a member of a surface that is well represented by the calculated coefficients. A value of one means that there is absolute confidence that the local surface is represented by the calculated coefficients; a value of zero means that there is no confidence in this surface - not that the surface is definitely incorrect.

4.3 Surface Similarity Beliefs

Given a pixel, p, and its neighbour, n, with their associated surfaces expressed in terms of the fitted biquadratic coefficients it is now necessary to decide on the likelihood of these surfaces being in fact the same and the two pixels therefore linked. The two sets of coefficients, C_p and C_n, are expressed relative to the centres, O_p and O_n, of their respective best fit windows as depicted in Figure 4-5. The positions of each pixel in

their best fit windows have been stored, (x_p,y_p), (x_n,y_n), and as the relative position of p and n is a matter of definition, being either (1,0), (0,1), (-1,0) or (0,-1), so the position of O_n relative to O_p can be found.

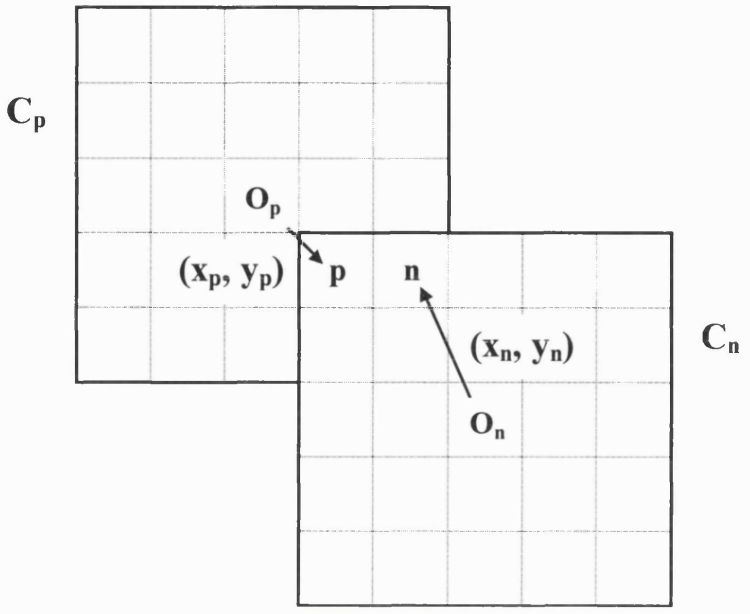


Figure 4-5 Pixel and neighbour coordinate systems

In order to compare the surfaces, the two sets of coefficients need to be expressed relative to the same origin. Transposing the coefficients of a surface from one coordinate system to another is achieved as follows:

If the surface is defined by coefficients c_{ij} in coordinate system (x,y) as,

$$f(x,y) = c_{00} + c_{10}x + c_{01}y + c_{11}xy + c_{20}x^2 + c_{02}y^2$$

then in another system, (x',y'), translated (x_T,y_T) relative to the first;

$$x' = x - x_T, \quad y' = y - y_T$$

the surface becomes;

$$\begin{aligned} f(x \to x', y \to y') = {} & c_{00} + c_{10}(x' + x_T) \\ & + c_{01}(y' + y_T) + c_{11}(x' + x_T)(y' + y_T) \\ & + c_{20}(x' + x_T)^2 + c_{02}(y' + y_T)^2 \end{aligned}$$

which can be rewritten,

$$
\begin{aligned}
f(x',y') = {} & c_{00} + c_{10}x_T + c_{01}y_T + c_{11}x_Ty_T + c_{20}{x_T}^2 + c_{02}{y_T}^2 \\
& +(c_{10} + c_{11}y_T + 2c_{20}x_T)x' \\
& +(c_{01} + c_{11}x_T + 2c_{02}y_T)y' \\
& +c_{11}x'y' \\
& +c_{20}x'^2 \\
& +c_{02}y'^2
\end{aligned}
$$

This is defined in terms of coefficients in the (x', y') system by;

$$f(x',y') = c'_{00} + c'_{10}x' + c'_{01}y' + c'_{11}x'y' + c'_{20}x'^2 + c'_{02}y'^2$$

and consequently the coefficients are related by;

$$
\begin{aligned}
c'_{00} &= c_{00} + c_{10}x_T + c_{01}y_T + c_{11}x_Ty_T + c_{20}{x_T}^2 + c_{02}{y_T}^2 \\
c'_{10} &= c_{10} + c_{11}y_T + 2c_{20}x_T \\
c'_{01} &= c_{01} + c_{11}x_T + 2c_{02}y_T \\
c'_{11} &= c_{11} \\
c'_{20} &= c_{20} \\
c'_{02} &= c_{02}
\end{aligned}
$$

from which it can be seen that whilst the second order coefficients are unchanged by the translation, the constant and first order terms *are* affected. Any error in the original coefficients will lead to possibly inflated inaccuracies in the translated values and this can be especially the case for c'_{00} if the translation vector, (x_T, y_T), is large. For this reason, in order to minimise the overall translation distances, the coefficients of the surfaces of both the pixel, p, and its neighbour, n, were translated to the midpoint of a line connecting the two origins of their separate coordinate systems.

Having obtained the two sets of coefficients, now expressed relative to the same coordinate system, the problem is now to compare them and obtain a belief value for their similarity. Initially this was done by using each set of translated coefficients to generate a new window of 25 pixel values. These two windows were differenced and the root mean square error found. The belief value was then calculated using a similar formula to that of Eq 4-11. However this seemed unnecessarily arbitrary in that any size of window could have been generated, with larger ones being more sensitive to

errors in the coefficients. A more direct formula for establishing the difference, δ_s, between the two surfaces, S_p and S_n, was therefore constructed which also gives control over the effective significance of the different orders of coefficients;

$$\delta_s = \alpha(|\delta_{00}|) + \beta(|\delta_{10}| + |\delta_{01}|) + \gamma(|\delta_{11}| + |\delta_{20}| + |\delta_{02}|) \qquad \text{Eq 4-12}$$

where, $\delta_{ij} = c_{ij} - c'_{ij}$, are the coefficient differences

The parameters, α,β,γ, are to be determined either automatically or by the user from a knowledge of the image noise content; this is discussed further in Section 4.7 and in Chapter 5.

This choice of formula for the detemination of δ_s is quite arbitrary and it might well be that a version based on the residual errors found by comparing two generated windows, as described earlier, would ultimately provide a more satisfactory method.

The belief of surface similarity is then computed as;

$$B(S_p = S_n) = \exp\left\{-\ln(2)\left(\frac{\delta_s}{\sigma}\right)^2\right\} \qquad \text{Eq 4-13}$$

and relies on the same parameter σ as does the belief of fitting function already described, Eq 4-11. The meaning ascribed to this function is as follows: a value of one denotes absolute confidence that the surfaces are exactly the same and a value of zero that they are very different.

4.4 Pixel Linking Beliefs

The previous two sections have described the method of obtaining both the belief, $B(p \in S_p)$, that a surface, S_p, as defined by the calculated coefficients, is a good representation of the local area including the pixel, p, and the belief, $B(S_p = S_n)$, that two surfaces, S_p and S_n, are the same. The belief that a pixel, p, and its neighbour, n, are linked, which is written $B(p \approx n)$, is defined here to be simply the product of these separate beliefs;

$$B(p \approx n) = B(p \in S_p)B(n \in S_n)B(S_p = S_n) \qquad \text{Eq 4-14}$$

The following Table 4-1, shows the values of $B(p \approx n)$ reached for all the possible combinations when the component beliefs have the values zero or one. It is important to realise that these are only the extreme cases, listed for convenience, and that belief functions in general will take intermediate values.

$B(p \in S_p)$	$B(n \in S_n)$	$B(S_p = S_n)$	$B(p \approx n)$
1	1	1	1
1	1	0	0*
1	0	1	0*
1	0	0	0
0	1	1	0*
0	1	0	0
0	0	1	0
0	0	0	0

Table 4-1 Beliefs of linking

The top row of the table shows the condition for certainty that the pixels are linked; the two local surfaces are perfectly fitted by their coefficients and these coefficients are exactly the same. All other values of $B(p \approx n)$ are zero which might be taken as an indication that the pixels are definitely not linked. There is a difficulty in interpreting all the resultant $B(p \approx n)$ values in this same way.

A certainty that p and n are not linked would imply that they are definitely members of different surfaces. This situation is only really the case if there is confidence that the two local surfaces are exact fits and they are very different. This is the case when S_p is a perfect fit, $B(p \in S_p) = 1$, S_n is a perfect fit, $B(n \in S_n) = 1$, and S_p and S_n are quite different, $B(S_p = S_n) = 0$ as given in the second row of the table. Table 4-1 shows other combinations of the component beliefs resulting in a zero value of $B(p \approx n)$ which are not certainties of non-linking; for example, if $B(p \in S_p)$ is zero then p might be a member of the same surface, S_n as n, or of some totally different surface

(irrespective of the values of $B(n \in S_n)$ and $B(S_p = S_n)$ and hence there is only uncertainty as to the linking. In general if either or both of $B(p \in S_p)$ and $B(n \in S_n)$ are zero then the resultant belief should represent uncertainty of linking/non-linking and not definite non-linking.

This can be expressed by the fact that 1- $B(p \approx n)$ is not the belief of the pixels p and n being not-linked. A belief of non-linking $B(p \not\approx n)$ can be constructed using the consideration given above.

For convenience a more convenient notation for the beliefs of fitting and surface similarity is first defined:

$$P = B(p \in S_p) \qquad \text{Eq 4-15}$$
$$N = B(n \in S_n)$$
$$S = B(S_p = S_n)$$

The belief of non-similarity of the surfaces is then given by;

$$\overline{S} = 1 - S = 1 - B(S_p = S_n) \qquad \text{Eq 4-16}$$

The belief of linking becomes,

$$B(p \approx n) = PNS \qquad \text{Eq 4-17}$$

and the belief of non-linking can be defined as;

$$B(p \not\approx n) = PN\overline{S} \qquad \text{Eq 4-18}$$

The inverses of P and N are defined;

$$\overline{P} = 1 - P = 1 - B(p \in S_p) \qquad \text{Eq 4-19}$$
$$\overline{N} = 1 - N = 1 - B(n \in S_n)$$

Because it is necessary that,

$$1 = (P + \overline{P})(N + \overline{N})(S + \overline{S})$$

it follows that

$$1 = PNS + \overline{P}NS + P\overline{N}S + PN\overline{S} + \overline{P}\,\overline{N}S + \overline{P}N\overline{S} + P\overline{N}\,\overline{S} + \overline{P}\,\overline{N}\,\overline{S}$$
$$= B(p \approx n) + B(p \not\approx n) + \overline{P}NS + P\overline{N}S + \overline{P}\,\overline{N}S + \overline{P}N\overline{S} + P\overline{N}\,\overline{S} + \overline{P}\,\overline{N}\,\overline{S}$$

and therefore $B(p \approx n) + B(p \not\approx n)$ does not sum to one as might at first have been expected. In fact;

$$B(p \approx n) + B(p \not\approx n) = 1 - (\overline{P}NS + P\overline{N}S + \overline{P}\,\overline{N}S + \overline{P}N\overline{S} + P\overline{N}\,\overline{S} + \overline{P}\,\overline{N}\,\overline{S})$$

and therefore;

$$B(p \approx n) + B(p \not\approx n) \leq 1$$

4.5 Comparison with Dempster-Schafer methods

The Dempster-Schafer belief system as described by Schafer[83] is partly an attempt to remove the restrictive nature of probability based methods. This is perhaps best demonstrated by an example. We postulate a proposition, X,

X: 'Tony Blair has a cold' .

Then a probabilistic approach would have as a basis that X is either true or false and therefore that the probabilities of its truth, Pr(X) and of its falsehood must sum to one:

$$\Pr(X) + \Pr(\overline{X}) = 1$$

But suppose that Joe Smith has never heard of Tony Blair. Then his belief (Bl) in the proposition is without evidence and therefore:

$$\mathrm{Bl}(X) = 0.$$

But equally he has no evidence to the contrary and therefore in addition;

$$\mathrm{Bl}(\overline{X}) = 0,$$

showing that beliefs do not obey the same rules as probabilities.

The Dempster-Schafer method defines another variable, the plausibility;

$$Pl(X) \equiv (1 - Bl(\overline{X}))$$

and constructs a belief range, [Bl(X), Pl(X)], the first part of which is interpreted as the current belief and the second, the plausibility, as the possible figure which the belief might reach given new evidence.

A whole formalism is developed for calculating these beliefs and for their recalculation as new evidence is added.

The important points for comparison with the system described here are firstly that a belief of zero does not imply the invalidity of the proposition but rather a lack of knowledge about it and secondly the separate independent definitions of the beliefs of truth and falsehood and consequent non-necessity of summation to unity.

The belief of pixel fitting, Eq 4-11, takes a value of one if there is absolute confidence that the local surface is that represented by the calculated coefficients. A zero value means that there is no confidence and *not* that the surface is necessarily incorrect. As such this belief corresponds well to that of Dempster and Schafer.

The belief of surface similarity, Eq 4-13, is more analogous to a probability in that it takes the values one and zero to mean the surfaces are respectively exactly the same or very different.

The beliefs of linking, Eq 4-17 and non-linking, Eq 4-18, take the value of one when there is absolute confidence but zero implies simply a lack of information. Furthermore these beliefs are defined separately and do not necessarily sum to one. Consequently they are analogous to the Dempster-Schafer beliefs.

No use of the Dempster-Schafer system has been made in this work and, indeed knowledge of it was not made until after the works completion.

4.6 Optimum Segmentation

Having obtained the beliefs of linking and of non-linking these are now used to define a segmentation or labelling. It is possible to define a quality or strength of any given labelling in terms of the belief values.

For convenience an abbreviated form of notation for the beliefs of connection and disconnection of a pixel, p, and one of its neighbours, n is adopted;

$$C_{pn} \equiv B(p \approx n)$$
$$D_{pn} \equiv B(p \not\approx n) \qquad \text{Eq 4-20}$$

Now if the belief values indicate that p and n are connected rather than disconnected, $C_{pn} > D_{pn}$, then it would be expected that the pixels would be more likely to have the same label than not. Given any belief set, it will not, in general, be possible to label all pairs of neighbouring pixels by such a simple criterion: chains of strong beliefs of disconnection do not always form closed boundaries and label ambiguities would occur (see chapter 2). However this idea of labelling compatibility with the beliefs enables us to define a strength of segmentation and further, the concept of the best segmentation as that with the maximum strength.

Following Chapter 2 segmentations or labellings are designated by the symbol l, and a particular segmentation by l_a so that a pixel, p, has a label $l_a(p)$. The neighbourhood set of the pixel p at (x_p, y_p) is defined as Q_p and consists, for four-neighbour connectivity, of the pixels at (x_p+1, y_p), (x_p-1, y_p), (x_p, y_p+1) , (x_p, y_p-1).

For any segmentation, l_a, the strength of segmentation, S_a, is defined as the sum over all pairs of neighbouring pixels, p and n, accumulating the belief of connection when the labels are the same and the belief of disconnection when they are different;

$$S_a \equiv \sum_{\substack{p \in I \\ l_a(p)=l_a(n)}} \sum_{n \in Q_p} C_{pn} + \sum_{\substack{p \in I \\ l_a(p) \neq l_a(n)}} \sum_{n \in Q_p} D_{pn} \qquad \text{Eq 4-21}$$

The first double summation in Eq 4-21 is for pixels and neighbours with the same label, that is pairs within regions and the second for those with different labels, that is pairs which straddle a region boundary. This definition quantifies the degree with which the chosen labelling is in agreement with the calculated connection and disconnection beliefs.

The best segmentation is defined as, l_{best}, where,

$$S_{best} \equiv \max\{S_a\}$$ Eq 4-22

The formula for strength of segmentation, Eq 4-21, can be rewritten in two useful ways by noting that the form of the two double summations are each part of another taken over the whole image without restriction on the labels;

$$\sum_{p\in I}\sum_{n\in Q_p} \equiv \sum_{\substack{p\in I \; n\in Q_p \\ l_a(p)=l_a(n)}}\sum + \sum_{\substack{p\in I \; n\in Q_p \\ l_a(p)\neq l_a(n)}}\sum$$ Eq 4-23

Using this, firstly to substitute for the summation of the C_{pn} in Eq 4-21 produces;

$$S_a \equiv \sum_{p\in I}\sum_{n\in Q_p} C_{pn} + \sum_{\substack{p\in I \; n\in Q_p \\ l_a(p)\neq l_a(n)}}\sum (D_{pn} - C_{pn})$$ Eq 4-24

and secondly for the summation of the D_{pn};

$$S_a \equiv \sum_{p\in I}\sum_{n\in Q_p} D_{pn} + \sum_{\substack{p\in I \; n\in Q_p \\ l_a(p)=l_a(n)}}\sum (C_{pn} - D_{pn})$$ Eq 4-25

The first double summation terms in both Eq 4-24 and Eq 4-25 are constants, independent of the labelling and play no part in the maximisation of the functions and therefore the best segmentation can be determined as the l_a which maximises either,

$$S'_a = \sum_{\substack{p\in I \; n\in Q_p \\ l_a(p)=l_a(n)}}\sum (C_{pn} - D_{pn})$$ Eq 4-26

or,

$$S''_a = \sum_{\substack{p\in I \; n\in Q_p \\ l_a(p)\neq l_a(n)}}\sum (D_{pn} - C_{pn})$$ Eq 4-27

Up to now the labelling can be taken in the general sense, as described in Chapter 1, with the number of labels being anything from four (four-colour map labelling) to R_a, the total number of regions associated with the segmentation l_a, (region-number labelling). As Eq 4-27 only considers pairs of pixels which straddle region boundaries it can be rewritten to sum over each boundary in turn;

$$S_a'' = \sum_{r=1}^{R_a} \sum_{s=1}^{R_a} \sum_{\substack{p \in I \\ l_a(p)=r}} \sum_{\substack{n \in Q_p \\ l_a(n)=s}} (D_{pn} - C_{pn}) \qquad \text{Eq 4-28}$$

providing that a restriction is now made to region-number labelling with the labels running from 1 to R_a. Further, to avoid the unnecessary inclusion of each boundary twice, the second summation in Eq 4-28 can be changed to include only those regions with labels greater than the first;

$$S_a''' = \sum_{r=1}^{R_a} \sum_{s=r+1}^{R_a} \sum_{\substack{p \in I \\ l_a(p)=r}} \sum_{\substack{n \in Q_p \\ l_a(n)=s}} (D_{pn} - C_{pn}) \qquad \text{Eq 4-29}$$

This can be written,

$$S_a''' = \sum_{r=1}^{R_a} \sum_{s=r+1}^{R_a} \Omega(r,s) \qquad \text{Eq 4-30}$$

where $\Omega(r,s)$ is the strength of the boundary between regions r and s;

$$\Omega(r,s) = \sum_{\substack{p \in I \\ l_a(p)=r}} \sum_{\substack{n \in Q_p \\ l_a(n)=s}} (D_{pn} - C_{pn}) \qquad \text{Eq 4-31}$$

It is noted that $\Omega(r,s)$ only has the possibility of being non-zero when r and s are neighbouring regions and that therefore the second summation in Eq 4-31 is effectively further restricted to s values which are neighbours of r as well as being greater than r.

From Eq 4-30 it is seen that the segmentation strength is the sum of the strengths of all the individual boundaries and it follows that a necessary condition for the existence of any particular boundary in the optimum segmentation is;

$$\Omega(r,s) > 0 \qquad \text{Eq 4-32}$$

as, if this were not the case, the segmentation strength could be increased by removing the boundary and merging regions r and s by re-labelling all s-labelled pixels to r or vice-versa. However it is important to realise that Eq 4-32 is not a sufficient condition for the existence of a boundary which is part of the optimum segmentation: many

boundaries will exist which satisfy this criterion but are not part of the maximum strength segmentation.

This condition can be used to show why the definition of an optimum segmentation given earlier, Eq 4-22, is satisfactory in that it fits a common sense idea of such a delineation. Consider a very simple image consisting of two areas of constant grey level as depicted in Figure 4-6. The beliefs of connection and disconnection are as given in the same figure. There are four belief values for each pixel but only those for a pixel to its right hand neighbour are shown: the vertical values are clearly not of interest for this image being either all +1 (connections) or -1 (disconnections).

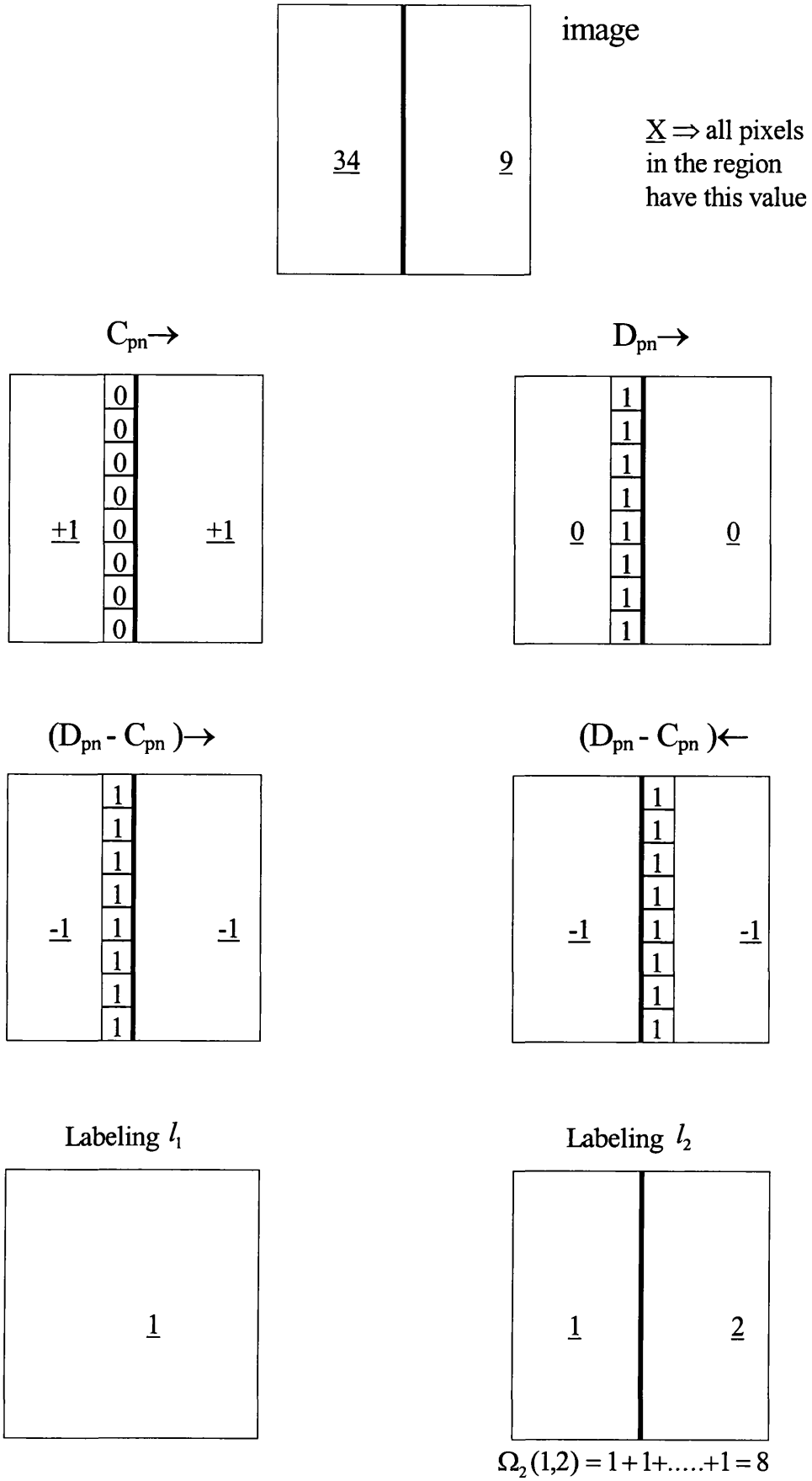


Figure 4-6 Beliefs for simple image

For this simple image a perfectly fitting surface can be found for all pixels and hence the error values for all pixels and the rms errors for all the chosen windows are zero. Consequently the belief of fitting values for all pixels are one. Furthermore, as all the surfaces on each side of the true boundary are exactly the same, the beliefs of surface similarity are one for pixels on the same side of the boundary. Only pixel-neighbour pairs straddling the boundary have different surfaces and by choosing the error parameter, σ, appropriately, the surface similarity beliefs for these pairs are calculated as zero. As a result all pixels have a belief of connection value of 1 except those in a vertical strip to the left of the true boundary which are zero. By similar reasoning the beliefs of disconnection are zero everywhere except in the same strip where they are one. The disconnection- connection differences are also shown and it is these that are summed in the formula for the boundary strength, Eq 4-21;

$$S_a \equiv \sum_{\substack{p \in I \\ l_a(p)=l_a(n)}} \sum_{n \in Q_p} C_{pn} + \sum_{\substack{p \in I \\ l_a(p) \neq l_a(n)}} \sum_{n \in Q_p} D_{pn}$$

Figure 4-6 also shows two possible segmentations designated l_1 and l_2 the second of which has a single boundary in the *correct* place. Applying the strength formula to this boundary the value $\Omega_2(1,2) = 8$ is obtained and as this is positive, the total strength of segmentation for l_2 is greater than that for l_1 showing that it is the preferable segmentation as is intuitively obvious.

However this would have been the case for all positive values of $\Omega_2(1,2)$. More realistic images contain boundaries which are not so clearly defined at every pixel. For example if the disconnection-connection differences were everywhere the same as Figure 4-6 except that;

instead of the strip,

1
1
1
1
1
1
1
1

there exists:

1
0.8
0.3
-0.4
-0.2
0.3
0.7
1

which will occur when the true region surfaces tend to have similar properties towards the centre, then the boundary strength would be reduced but still positive; $\Omega_2(1,2) = 3.5$, and the two-region labelling still preferred. As the region surfaces become more similar so more negative disconnection-connection values would be calculated and eventually the total boundary strength would become negative and the single region labelling chosen.

4.7 Adjustment of the fitting belief parameters

The segmentation, as defined in terms of the beliefs of disconnection and connection (Eq 4-22, Eq 4-30, Eq 4-31) is dependent on the beliefs of pixel fitting and surface similarity (Eq 4-11, Eq 4-13) and these are themselves dependent on the values of the 'noise' parameter σ, and those parameters, α, β, γ, which are part of the definition of the surface difference (Eq 4-12). These parameters have to be set either by the user or perhaps in some future version automatically. It is needed to gain some idea of the effect of these parameters and define some method of deciding how 'correct' values can be chosen.

Our requirement is twofold; firstly that smooth surfaces be segmented as a single region; this indicates that the $D_{pn} - C_{pn}$ values should be mainly negative in such a region. Secondly, for pixel-neighbour pairs straddling a boundary between two surfaces the $D_{pn} - C_{pn}$ are required to be largely positive, that is;

- on a 'smooth' surface: $D_{pn} - C_{pn} \lesssim 0$ Eq 4-33
- at the boundary of surfaces: $D_{pn} - C_{pn} \gtrsim 0$

It is further required that these conditions are strengthened when the confidence in the calculated surface coefficients is high. The effect of these requirements is examined by rewriting the $D_{pn} - C_{pn}$ in terms of the beliefs of pixel and neighbour fittings and the belief of surface similarity, P, N, S, respectively and the definitions of their inverses Eq 4-17 and Eq 4-19. Using Eq 4-17, Eq 4-18, and Eq 4-20 yields;

$$D_{pn} - C_{pn} = PN\bar{S} - PNS = PN(\bar{S} - S) = PN(1 - 2S) \quad \text{Eq 4-34}$$

and if the pixel and neighbour fitting are taken to be the same (only worst cases can be considered so this is not a loss of generality);

$P = N = B$ (say)

then;

$$D_{pn} - C_{pn} = B^2(1-2S) \qquad \text{Eq 4-35}$$

from which the following table can be constructed:

B \ S	0	1/4	1/2	3/4	1
0	0	0	0	0	0
1/4	1/16	1/32	0	-1/32	-1/16
1/2	1/4	1/8	0	-1/8	-1/4
3/4	9/16	9/32	0	-9/32	-9/16
1	1	1/2	0	-1/2	-1

Table 4-2 $D_{pn} - C_{pn}$ values

The table shows how pixels p and n are only clearly designated linked or unlinked when the fitting beliefs (B) are high ($\rightarrow$1). Linked pixels ($D_{pn} - C_{pn} \gtrsim 0$) arise from a high surface similarity belief (S$\rightarrow$1) and unlinked ($D_{pn} - C_{pn} \lesssim 0$) from a low value (S$\rightarrow$0).

If within a 'true' surface the maximum root mean square error occurring is $\overline{\Delta}_{rms}$, say, (see later for examples of causes) then from the above considerations the fitting beliefs might be imposed to have a minimum value of about 0.5 at such places (and hence higher values elsewhere). This implies, using Eq 4-11, that σ should be set such that;

$$(\overline{\Delta}_{rms} / \sigma) \lesssim 1 \quad \text{or} \quad \sigma \gtrsim \overline{\Delta}_{rms} \qquad \text{Eq 4-36}$$

Setting σ a lot higher than $\overline{\Delta}_{rms}$ will mean that all the P, N and S values will tend to their maximum of 1 and all pixels be linked irrespective of any surface differences.

In addition, from the first of the conditions Eq 4-33, it can be seen that for pixel-neighbour pairs on a 'true' surface it is required that $S \gtrsim 0.5$ which, if the maximum 'natural' difference errors occurring (see later) are $\overline{\delta}_s$ using Eq 4-13, implies:

$$\frac{\overline{\delta}_s}{\sigma} \lesssim 1 \quad \text{or} \quad \overline{\delta}_s \lesssim \sigma \qquad \text{Eq 4-37}$$

This puts a restriction on the different surfaces that can be distinguished; they must have actual surface differences greater than this. Because of the definition of δ_s, Eq 4-12, requirement Eq 4-37 implies conditions on α, β, and γ.

By considering the fitting and surface difference errors that occur in a particular image the conditions, Eq 4-36 and Eq 4-37 can be applied to calculate suitable values for these adjustable parameters.

Firstly, and most simply, consider an image consisting only of perfect biquadratic surfaces. All the fitting errors, Δ_{rms}, will be zero and the local coefficients an exact copy of the *true* coefficients. The first requirement, Eq 4-36, will be then be satisfied providing only that σ is non-zero. For pixel-neighbour pairs on the same surface the local surface coefficients will be exactly the same and hence the δ_s will be zero and the second requirement, Eq 4-37, satisfied, again providing only that σ is non-zero. For pixel-neighbour pairs straddling a true boundary the local coefficients will, by definition, be different and by making α, β, and γ very large and σ small, arbitrarily small differences in real surfaces can be found.

To summarise, if it were known that the surfaces were perfect biquadratics then the σ could be set to be small, α, β, γ to be large and surfaces with arbitrarily small differences in their true coefficients segmented as different regions,.

However, most images are not composed of perfect biquadratic surfaces. Variations arise because of four main causes:

1. The changes arising when only a finite number of grey levels are used to represent an image. For example, a one byte image with 256 grey levels set to 0, 1, 2, ...255, can only approximate the grey level values of a simple ramp with gradient 0.5 (say): the true values; 0, 0.5, 1, 1.5, 2, 2.5,... will become; 0, 0, 1, 1, 2, 2,... which is no longer an exact biquadratic. These digitisation errors mean that usually even an originally exact biquadratic surface will have only an approximate fitting.

2. Most smooth surfaces are not exact biquadratics and will only be approximated by such an expression.

3. The noise present in many images means that even in the absence of the above two variations a biquadratic will not be an exact fit.

4. The finite window size used means that for some surfaces there will exist pixels for which it is not possible to site a window which both contains the pixel and is fully covered by the surface. This is self-evidently true for regions which have a dimension smaller than that of the window (5) but also occurs when the boundary of the surface has a convexity such that it will not allow of a window including a pixel without at the same time intersecting the boundary. Such windows will straddle true surface boundaries and, even in an image of perfect biquadratics, will still contain fitting errors.

It is instructive to obtain some measure of the typical errors in fitting and the surface difference values that occur for single surfaces as produced by causes 1 and 2 above.

Table 4-3 shows some values obtained. The following points of explanation are needed.

- All the images were generated initially with exact values and then rounded down to the nearest integer and formed to a one-byte image

- All, apart from the last two surfaces, are biquadratics simulated using just one coefficient; either the first or second order in one direction.

- The maximum values of c_{10} and c_{20} are chosen to give the roughly the greatest possible spread of pixel values (0 to 255) over a relatively small dimension, chosen to be 16 pixels, and therefore represent quite a severe test.

Surface type	Max Δ_{rms}	max $\lvert\delta_{00}\rvert$	max $\lvert\delta_{10}\rvert$	max $\lvert\delta_{20}\rvert$
Biquad, c_{10}=0.18	0	1	0	0
Biquad, c_{10}=0.40	0.585	0.479	0.629	0.143
Biquad, c_{10}=9.40	0.557	0.439	0.575	0.143
Biquad, c_{10}=15.7	0.585	0.843	1.143	0.286
Biquad, c_{20}=0.24	0.788	0.853	0.714	0.076
Biquad, c_{20}=0.51	0.557	0.842	1.143	0.286
Biquad, c_{20}=0.73	0.294	0.325	0.543	0.214
Biquad, c_{20}=0.99	0.408	0.842	0.628	0.143
Exp (0.3x)	0.796	1.004	2.89	0.796
Exp (0.36x)	0.867	3.14	3.14	2.26

Table 4-3 Typical fitting and surface difference errors

- The first surface has a gradient of 0.18. As this is less than 0.2, the image, after quantisation, has sequences of at least 5 pixels of the same value; 1,1,1,1,1,2,2,2,2,2,... Consequently the fittings (using a 5 by 5 window) are exact – to constant value surfaces – with the c_{10} and c_{20} coefficients calculated as zero.

- The second column of the table gives the maximum root mean square error in fitting, Δ_{rms}. This is the maximum of the errors of the windows *which are used.*

- The third to fifth columns give the maximum absolute surface coefficient differences as used in the calculation of the total surface difference Eq 4-12. As the only variation in the images used is in one direction the other differences are all zero and hence is not necessary to show them.

As can be seen the maximum root mean square errors are fairly consistent and all under about 0.87 and this would seem to be the case for the digitisation type errors for all smooth surfaces.

For these surfaces by applying Eq 4-36, and setting $\sigma = 1$ (say) all the beliefs of fitting would be expected to be acceptably high within the surfaces. The worst case δ_{ij} surface differences occurred for the exponential and in fact these high values happened, as might, be expected, at the high-value/high-gradient end of the function. In order to produce linked pixel-neighbour pairs it is necessary to apply condition Eq 4-37 and use Eq 4-12, to set the α, β and γ such that:

$3.14\alpha + 3.14\beta + 2.26\gamma < \sigma\ (= 1)$

This could be achieved by setting $\alpha = 1/(3*3.14) \sim 0.1$ with similar evaluations for β and γ.

The above analysis gives a method of gaining some idea for appropriate settings of the parameters σ, α, β and γ but it must be stressed that other factors must be taken into consideration:

- It was assumed that the max δ_{ij} values for different ij, occur at the same pixel-neighbour; this is not necessarily the case.
- The user might be prepared to allow some real regions to be split providing that a higher proportion of true boundaries are found.
- Noise present in the image will increase both the $\overline{\Delta}_{rms}$ and δ_{ij} values. This can be countered by increasing σ.
- The method of producing a segmentation from the linked sets is not fixed. As will shown in Chapter 5, a simple thresholding method works but boundaries which are incomplete will be lost. This can be countered by increasing the α, β and γ to produce more non-links.

- It might be known that the surfaces present do not have such severe gradients and hence the maximum δ_{ij} will be lower. Again the α, β and γ could be raised in order to detect lower 'true' surface differences.

Obtaining the best segmentation for a given set of beliefs is a difficult task. The straightforward way would be to examine all possible segmentations and compute the strength of each one. However, as shown in Chapter 2, there are an enormous number of possible segmentations even for relatively small images and evaluating the strength of all of them is computationally impractical. In the next chapter a simple method of obtaining an approximate solution is demonstrated and in Chapter 6 optimisation methods as a way of realising the best segmentation are examined.

4.8 Summary

- A local surface fitting method using a biquadratic function in 5 by 5 windows was defined. For each pixel the best window was chosen as that with the lowest rms error of fitting.

- The mathematics of the fitting method were explained showing that the coefficients of the fitting are given by the product of a 'pseudo-matrix' constructed from the coordinates of pixels within the 5 by 5 window with a vector of the pixel grey levels.

- The outputs of the fitting process were demonstrated with a simple image consisting of biquadratic surfaces

- The 'belief' of fitting was defined for the best surface for each pixel as an exponential function of the rms error giving values between 0 (poor fitting) and 1 (perfect fitting).

- The belief of surface similarity for the surfaces of each pixel-neighbour pair was defined. Again an exponential function was used with the exponent this time being a weighted linear sum of the differences between the coefficients of fitting. The coefficients have first to be translated to the same coordinate axes.

- The beliefs of pixel-neighbour connection and disconnection were calculated in terms of the previous beliefs.

- The strength of segmentation was defined as a sum over all pixel-neighbour pairs of the beliefs of either connection or disconnection; when the pixel and neighbour labels are the same the belief of connection is included and when they are different the belief of disconnection. The optimum segmentation is then defined as that with the highest strength.

- Various forms of the optimum segmentation definition were given including one in the form of a summation of the boundary strengths of all neighbouring regions.

- A simple schematic demonstration was given to show how the mathematical definition of the best segmentation parallels the intuitive idea of the same by completing partial boundaries providing they are strong enough.

- An analysis of the effect of the user-defined parameters was made. The different ways in which real images can differ from ideal biquadratic surfaces was tabulated, some statistics for the errors so caused were recorded and the effect on the best choice of parameter values analysed.

5. Thresholded Belief Results

5.1 Thresholding Beliefs

Reiterating the findings of Chapter 4 that, given the beliefs of connection and disconnection of pixels, p, and their neighbours, n; C_{pn} and D_{pn} respectively, the best segmentation, l_a, is defined as that which maximises the strength function;

$$S_a'''=\sum_{r=1}^{R_a}\sum_{s=r+1}^{R_a}\Omega(r,s) \qquad \text{Eq 5-38}$$

where $\Omega(r,s)$ is the boundary strength between regions r and s;

$$\Omega(r,s)=\sum_{\substack{p\in I \\ l_a(p)=r}}\sum_{\substack{n\in Q_p \\ l_a(n)=s}}(D_{pn}-C_{pn}) \qquad \text{Eq 5-39}$$

The requirement, shown in Chapter 4, that,

$$\Omega(r,s)>0 \qquad \text{Eq 5-40}$$

is clearly easily satisfied if for all the pixel-neighbour pairs straddling the r, s boundary, $D_{pn}-C_{pn}>0$. It is therefore of interest to define a more restrictive segmentation which consists of boundaries for which not just the sum $\Omega(r,s)>0$ but for all the individual values, $D_{pn}-C_{pn}>0$.

This is most easily achieved by first defining an intermediate step in the process of obtaining the discrete label values from the belief sets which have continuous variation from 0 to 1. This is done using the concept of links between neighbouring pixels which have been previously described in Chapter 2. Links either take the value 0, meaning the pair are not-linked, or 1, meaning they are. Clearly pixel linking is strongly related to the beliefs of connection/disconnection and, as shown before, there is a direct relationship between linking and labelling.

Therefore a set of links, $L(p,n)$, is defined in terms of simple thresholding of the sets of beliefs of connection and disconnection, C_{pn} and D_{pn};

$$D_{pn} - C_{pn} < 0 \Rightarrow L(p,n) = 1$$
$$D_{pn} - C_{pn} \geq 0 \Rightarrow L(p,n) = 0 \qquad \text{Eq 5-41}$$

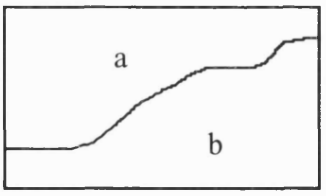
In general the links defined in this way will not be a self-consistent set but will contain the anomalies described in Chapter 2. These links can then be used to produce a segmentation by the simple procedure of collecting together linked groups of pixels as previously described in Chapter 2. This is termed a *zero-thresholded belief* segmentation.

Intuitively this *zero-thresholded belief* segmentation forms a subset of the *best* segmentation containing some, but not necessarily all, of the same boundaries and not containing any others. It is instructive to show that this is indeed the case. Starting with a definition:

A segmentation, l_1, is a subset of another, l_2, if all pairs of pixels which are labelled differently in l_1 are also labelled differently in l_2, as then all boundaries that exist in l_1 are also present in l_2.

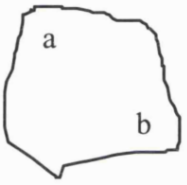
The proof then proceeds as follows:

- Consider any two pixels a and b labelled differently in the zero-thresholded belief segmentation, l_1. Separating these two pixels must be at least one continuous chain of transverse (straddling) pixel-neighbour pairs, (p,n) for all of which $D_{pn} - C_{pn} > 0$ and which divides the whole image into two parts each containing one of the two pixels.

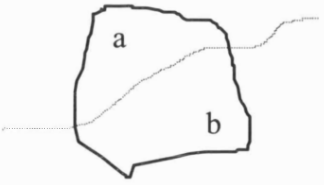


This must be the case as otherwise there would exist a longitudinal chain of pixels, with a and b as end-points, for which $D_{pn} - C_{pn} < 0$ for all p,n neighbouring pairs and this would necessitate a and b having the same label.

- The proof is by contradiction (*reductio ad absurdum*): the opposite of what it is sought to prove is assumed. Suppose that in the *best* segmentation, l_2, a and b are in the same region and therefore have the same label:



- Then part of the aforementioned chain must both bisect this region *and* cannot be part of l_2:



- But this part of the chain, like all of it, by definition, consists of straddling p,n pairs for all of which, $D_{pn} - C_{pn} > 0$, and hence the segmentation strength of l_2 could be increased by including this part and labelling the pixels on either side differently - that is by splitting this region.

- However the strength of the *best* segmentation, by definition, cannot be increased and hence the assumption that a and b are in the same region in l_2 must be incorrect.

- Consequently any two pixels in different regions in the zero-thresholded segmentation must also be in different regions in the best segmentation and therefore the former is a subset of the latter which is what was set out to be proved.

5.2 Over-Thresholding

In the original definition of segmentation strength, Eq 5-38, the actual values of the $D_{pn} - C_{pn}$ have a significant effect on the best segmentation: $D_{pn} - C_{pn} \to -1$ implies probable linking, $D_{pn} - C_{pn} \approx 0$ is ineffective and $D_{pn} - C_{pn} \to +1$ implies probable non-linking. However zero-thresholding means that all negative $D_{pn} - C_{pn}$ result in linked pixels irrespective of their actual value; only the sign of the $D_{pn} - C_{pn}$ is significant, the magnitude of no consequence. If the expression for $D_{pn} - C_{pn}$ in terms of the separate beliefs (4-24) is recalled:

$$D_{pn} - C_{pn} = PN(1-2S)$$

This means that the original beliefs of fitting based on the rms errors (P and N) do not matter in zero-thresholding as they do not affect the sign of the $D_{pn} - C_{pn}$. Only the belief of surface similarity matters; pixels are directly linked only if S > 0.5 (but can still be indirectly linked if an alternative chain of links is created).

It is, of course, possible to use a non-zero threshold. Instead of Eq 5-41 it is possible to define:

$$D_{pn} - C_{pn} < -T \Rightarrow L(p,n) = 1$$
$$D_{pn} - C_{pn} \geq -T \Rightarrow L(p,n) = 0$$

Choosing T negative serves little purpose as this would increase the number of linked pixels (over that of zero-thresholding) and hence reduce the number of regions from what is already a subset of the best segmentation. But choosing T positive reduces the amount of linking and hence increases the number of region boundaries formed. In *over-thresholding* the boundaries include all of the zero-thresholded subset of the ideal segmentation as well as some additional ones which, whilst they might not be exact components of the ideal segmentation, might nonetheless be an improvement in that

the strength of segmentation is increased by their presence. A consequence of non-zero thresholding is that the beliefs of fitting, P and N, are now important in determining which pixels are linked.

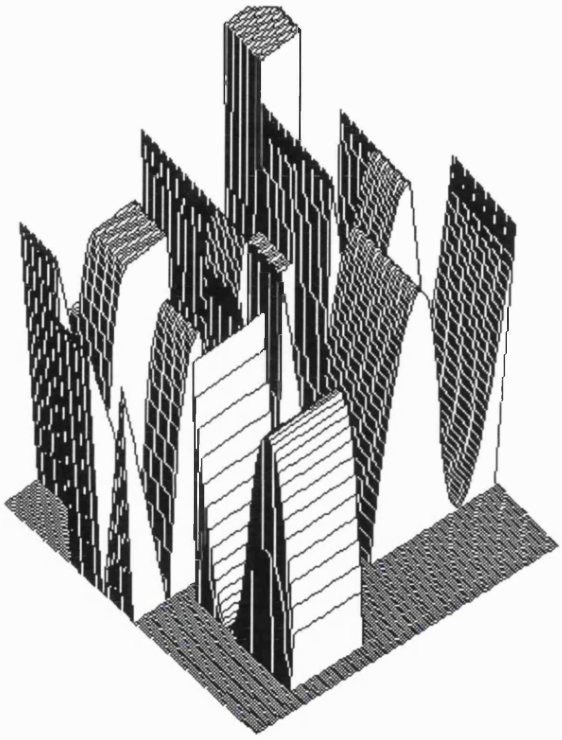
Choosing a very high value of T ($T \rightarrow +1$) would result in much reduced linking and far too many regions; ultimately each pixel would not be linked to any of its neighbours and would form a separate region. Evidently the choice of T together with that of the other parameters σ, α, β and γ is of importance in deciding the final segmentation.

The results presented in the remainder of this chapter are a demonstration of what can be obtained using this simple method of thresholding the beliefs.

5.3 Perfect quadratic surfaces

The first test image used was simulated with one byte per pixel (grey levels 0 to 255). It contains only perfect biquadratic surfaces including constants, ramps and quadratics. Figure 5-1 shows a three-dimensional illustration of the image and Figure 5-2 the image itself with the segmentation results. The image consists in the upper part of 16 by 16 pixel squares of varying surfaces and there are four 32 by 32 pixel regions in the lower part. The surface junctions were chosen to give a variety of tests for the surface coefficient difference comparator.

Figure 5-1, Test image with biquadratic surfaces shown in 3-D



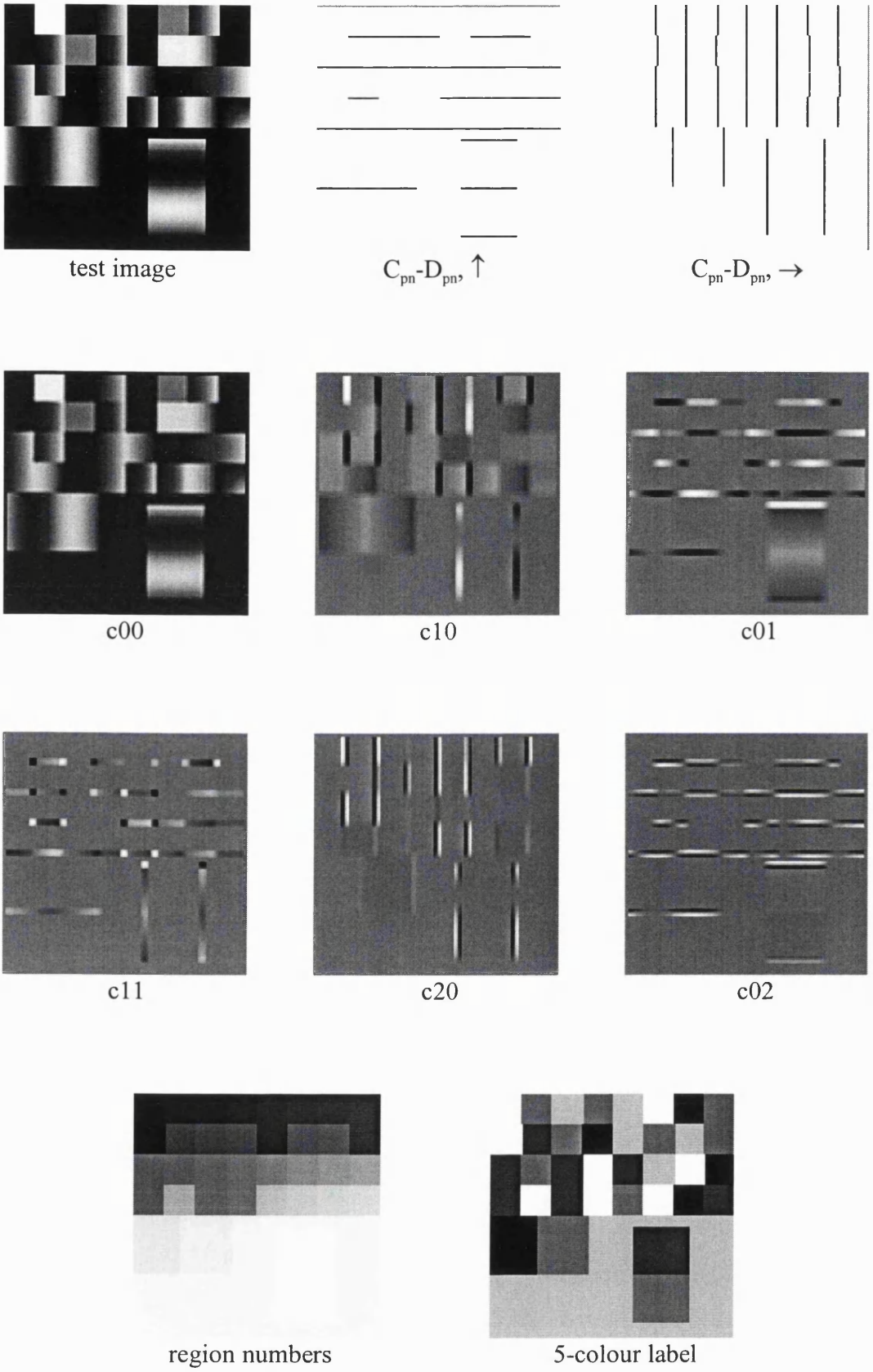


Figure 5-2, Test image results with $\sigma = 0.5$, $\alpha = \beta = \gamma = 2$

The six biquadratic coefficients are shown for all window positions. That for c_{00} shows a smoothed version of the original image which is effectively what it is. Those for c_{10} and c_{01} are equivalent to first order gradient operators in the horizontal and vertical directions respectively and show typical positive and negative peaks at the junctions of surfaces. The second order coefficients, c_{20}, c_{02} and c_{11} show the characteristics of second order derivative edge detectors with a typical zero crossing at each surface junction. It is important to realise that these coefficients are not all used; the coefficients associated with each pixel are those for its best fitting window and the coefficients calculated for badly fitting windows straddling a surface boundary are not utilised.

Figure 5-2 also shows the differences of the beliefs of connection and disconnection in the two directions they are calculated. For convenience these are shown in the negative of the form used in the definition of segmentation strength. Owing to the relative simplicity of this image these beliefs form almost a binary image with most of them being +1 indicating strong connectivity within regions and only those straddling boundaries being -1. Exceptions include the top line and right line of the upward and rightward direction beliefs respectively. These are zero arising from the fact that, as there is no pixel further up or to the right, the beliefs of both connectivity and disconnectivity are zero.

These belief differences are thresholded to produce linked and unlinked pairs of neighbouring pixels. Starting at the top left of the image, groups of linked pixels are given the same label to produce a segmentation in terms of region numbers which is also shown in Figure 5-2. This is not a particularly easy image to view the region boundaries as these tend to have grey level differences of just one and hence the regions can be hard to distinguish. The last image in Figure 5-2 shows a re-labelling of the region numbers using just 5 labels which for neighbouring regions are always chosen to be different thereby producing a much clearer image (the reasons behind 5 colour labelling will be explained in Chapter 6).

As can be seen a perfect segmentation was produced and this was done with a choice of segmentation parameters using the analysis given in section 3.6. Knowing the surfaces to be perfect it was possible to choose σ to be arbitrarily low and α, β and γ

arbitrarily high and in fact values of $\sigma = 0.5$, $\alpha = \beta = \gamma = 2$, enabled the result shown in Figure 5-2. All 31 regions were found with these values.

These parameter values would be somewhat impractical with a real image and Figure 5-3 shows equivalent results using the same test image with a parameter set of $\sigma = 2$, α, β and $\gamma = 0.3$. Examination of the beliefs of connection/disconnection differences shows that now there are some gaps where previously the boundary was continuous. Although the overall belief sets are clearly very similar, nonetheless these small gaps are sufficient to produce the different segmentations which comparison of the five-colour labellings in Figure 5-2 and Figure 5-3 shows. Regions have been lost, most notably two of the 32 by 32 quadratics in the lower part of the image have disappeared with the total reduced from 31 to 22 and this is a result of just one or two pixels being linked to their neighbours in the adjoining region, where previously they were not linked and the two regions now labelled as one.

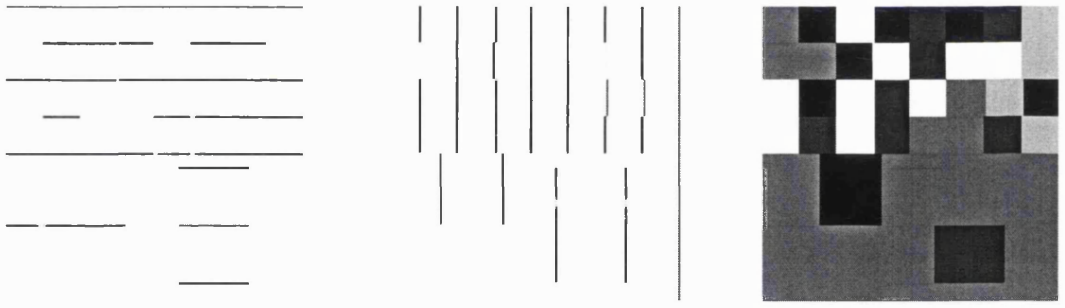


Figure 5-3 Test image results with $\sigma = 2$, α, β and $\gamma = 0.3$

This problem with the thresholding method is very frequent and will be encountered in future work.

5.4 Partially covered windows

In the previous example, see Figure 5-2, as all the regions are aligned with the sides of the image, for each pixel there exists at least one fitting window that both includes the pixel and also is wholly covered by the true surface containing the pixel. This resulted

in all pixels being in at least one perfectly fitting window. Figure 5-4 shows what happens when this idealised situation does not always hold. The image of 25^2 pixels consists simply of a square of bright pixels rotated 45° to the image dark background.

image fitting beliefs C_{pn}-D_{pn}, → 5 colour label

Figure 5-4 Effect of incompletely covered fitting window

In this image it is impossible for a window to cover pixels in the corners of the bright square without also including pixels from the background. As a consequence these corner pixels have very low fitting beliefs: the rms error of fitting is relatively high; this can be seen in the image of these fitting beliefs which everywhere are the maximum of +1 apart from these corners. The differences of beliefs of connectivity/disconnectivity are high everywhere apart from on the genuine boundaries and also within a small block of pixels in these corners. The segmentation shows that the main square is reasonably well delineated as one region but that the corners are split into a number of single pixel regions. The following table shows the actual region numbers for a block of pixels covering the bottom corner of the square in which the background has been labelled 1 and the square 8.

8	8	8	8	8	8	8	8
1	8	8	8	8	8	8	1
1	1	21	22	23	24	1	1
1	1	1	25	26	1	1	1
1	1	1	1	1	1	1	1

In fact each corner contains 6 single pixel regions which with the remainder of the bright square and the background gives a total of 26 instead of the ideal 2. This problem is clearly a result of the finite window size which is used. No attempt is made here to overcome this difficulty but suggestions for its avoidance are made in the final chapter.

5.5 Biquadratic surfaces quantised from real values

The images used throughout this study consist of one byte per pixel data with grey levels from 0 to 255. This restriction means that, even surfaces that were originally perfect quadratics will, when truncated to the one byte form, not necessarily remain so. As a result such surfaces will not present perfect fitting; the beliefs of fitting will not be precisely +1and neighbouring pixels will not be exactly linked. If such surfaces are to be correctly segmented as one region then the program parameters; σ, α, β, γ, have to be chosen suitably.

Consider first one of the simplest possible images divided into two halves each constant and separated in grey level by just one. All pixels will be perfectly fitted (P=N=1) and the associated first and second order coefficients will all be zero. The zeroth-order coefficient difference will be given by (see Chapter 4, Eq 4-3)

$$\left|c_{00} - c'_{00}\right| = 1$$

and pixel-neighbour pairs straddling the surface boundary will have a surface difference error of,

$$\delta_s = \alpha\left(\left|c_{00} - c'_{00}\right|\right) = \alpha$$

Therefore the surface similarity belief, S (see Chap 4, eq 4-4) becomes;

$$S = B(S_p = S_n) = \exp\left\{-\ln(2)\left(\frac{\delta_s}{\sigma}\right)^2\right\} = \exp\left\{-\ln(2)\left(\frac{\alpha}{\sigma}\right)^2\right\}$$

and setting, $\alpha/\sigma > 1$

then, $S < 0.5$

and the difference of beliefs of connection/disconnection (see Chapter 4, Eqs 4-24 and 4-25) are given by;

$$D_{pn} - C_{pn} = PN(1-2S) > 0$$

Therefore if using zero-thresholding the two surfaces will be separated as two regions. If α/σ is set <1 then separation could still be achieved using over-thresholding.

Now consider a simple ramp image arising from grey level gradients less than 0.2. In the original surface it will take 5 or more pixels before the grey level changes by at least 1. In the quantised image this will produce a series of strips with a width of 5 or more pixels of the same value. The program fittings will therefore be perfect but to constant surfaces as in the example above of the image split into two halves.

In this situation it might be thought desirable that the image is segmented as one region as this, by definition, is what the original surface consisted of. To achieve this using zero-thresholding would necessitate setting $\alpha/\sigma < 1$ the opposite of the requirement for the two surface image described previously.

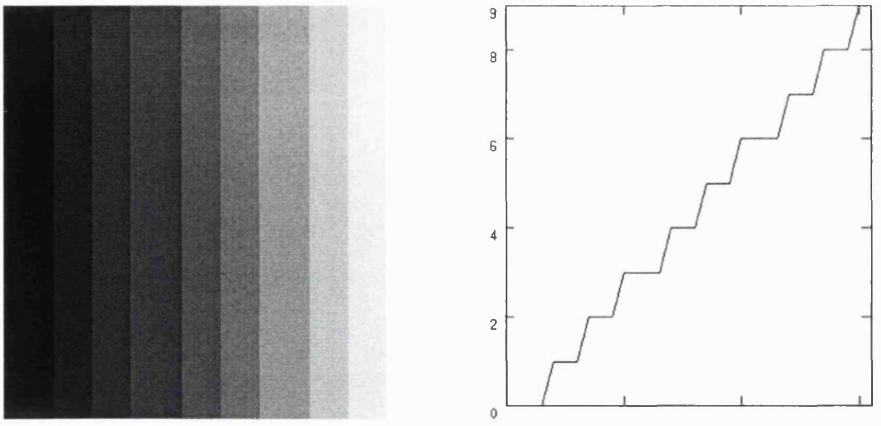
These two simple examples show the difficulty of achieving at the same time what initially appear to be two very simple requirements; segmenting as different regions constant surfaces separated by one grey level and as one region, a surface with a low gradient. This problem is apparent even with such artificially simple image types and is increasingly the case when more realistic images are considered. This highlights what must be appreciated in segmentation and which has already been discussed in chapter 1: the *correct* scene-segmentation solution is dependent on contextual information whereas the preliminary image-segmentation often will result in regions not corresponding to real objects.

Now consider ramp images in which the original gradient is greater than 0.2. After integisation this results in a series of steps some of which are less than 5 pixels in depth. The 5-pixel window fittings can now no longer be perfect and consequently all the coefficients take non-zero values; the root mean square fitting errors are non-zero

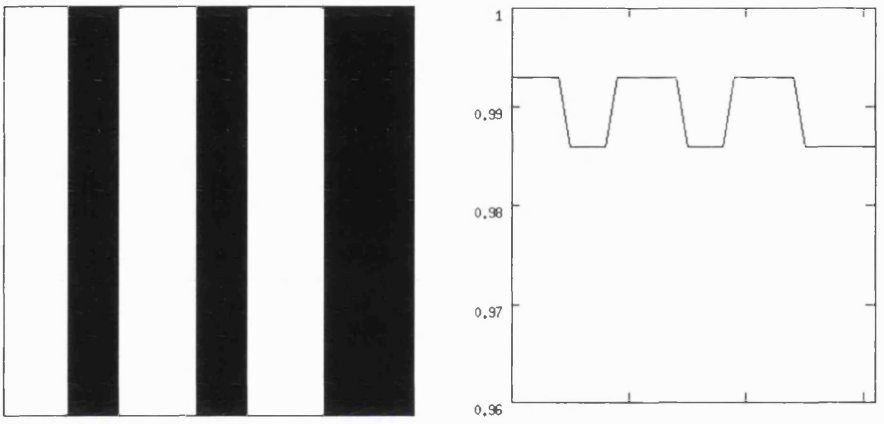
and in general all the surface coefficient differences non-zero. The surface similarity error therefore depends on the values of the β and γ parameters as well as σ and α.

As an example the image in Figure 5-5 is a quantised version of an original one of 32^2 pixels simulated as a simple ramp with a gradient of 0.3 in the horizontal direction and then integised to a one-byte image. The steps in grey level can be clearly seen. The parameters used were as shown and were selected simply with a view to demonstrating a typical result rather than in any sense an ideal one.

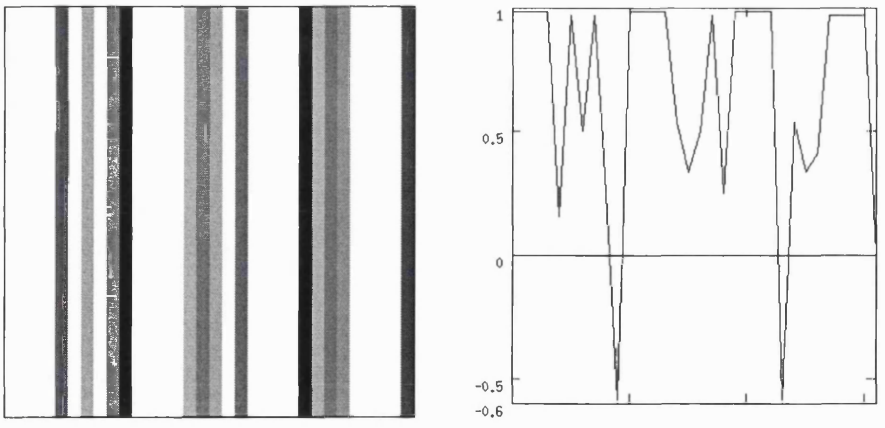
The belief of fitting values are relatively high, all of them being over 0.98. The distinct uniformity in the variation of the belief of fitting is a direct result of the pattern in the grey levels; starting at the left the image consists of four pixels with the same grey level of zero followed by three with the value one and then three at two. This pattern of 4-3-3 is then repeated across the rest of the image. The result of this uniformity is that the five by five pixel fitting windows only have a limited number of pixel grey level combinations.



Grey levels



Beliefs of fitting



C_{pn} -D_{pn}

Figure 5-5, Ramp results (c_{10}=0.3), σ=1.5, α=β=γ=1)

Numbering the pixels 0,1,2,3,..., in columns from the left, it can be seen that pixels numbered zero can only be in windows with four columns at grey level zero followed by one column at one; a 4-1 combination. Pixels numbered 1 and 2 are in both 4-1 and 3-2 combinations, pixels 3 and 4 in 4-1, 3-2 and 1-3-1 combinations and pixels 5,6,7,8 in 1-3-1 and 3-2 combinations. (Note that for the purposes of the fitting, the ordering of the column combination is irrelevant; 3-2 has exactly the same fitting accuracy as 2-3.) It happens that the arrangement with the most accurate fitting of these three types is the 4-1 combination and consequently this is chosen for pixels 0,1,2,3,4. Pixels 5,6,7,8 have only a (slightly) less accurate combination available to them and this pattern then repeats as shown in the graph; 6 pixels in a high accuracy window followed by four in a reduced accuracy one. The pattern is broken at the right edge of the image where the 4-1 combination is no longer available.

The differenced beliefs of connection/disconnection do not show quite the same uniformity. The reason for this lies in the way the surface similarity belief is calculated. Some neighbouring pixels are confined to have the same fitting window; for example pixels 0, 1, 2, 3, 4 all have just one 4-1 combination window available to them - that situated at the left-most position of the image - and therefore their surface coefficients will be the same and their surface similarity belief equal to one. However for many pixels (for example pixels 5, 6, 7, 8) there is a choice of windows with the same fitting accuracy but in different positions. It follows that different pairs of neigbouring pixels which appear to be in exactly comparable positions will have exactly the same window *combinations* allocated but not necessarily exactly the same window *positions*. As the values of the calculated coefficients depend very much on the translation vector (see Chapter 4 for details) the surface similarity beliefs for these pairs can well be different.

Although most values of the C_{pn}-D_{pn} are reasonably high (>0.6), in two places a negative value is seen and if zero-thresholding is used then a three region segmentation is produced as shown in the following image, Figure 5-6:

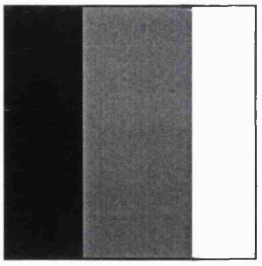
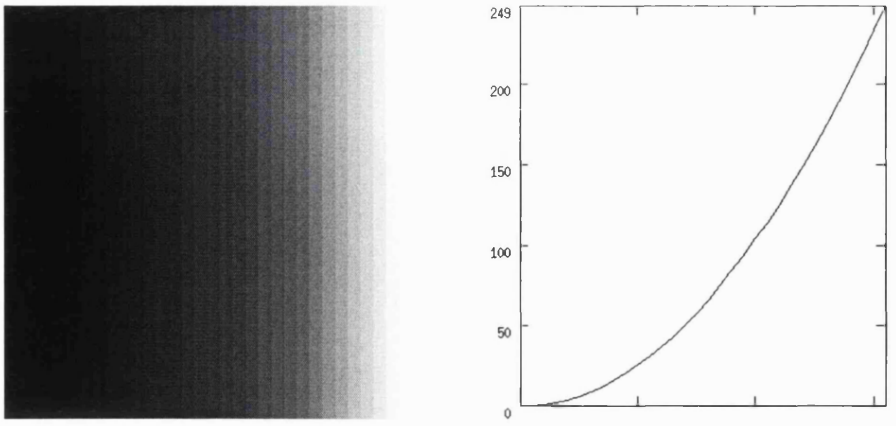


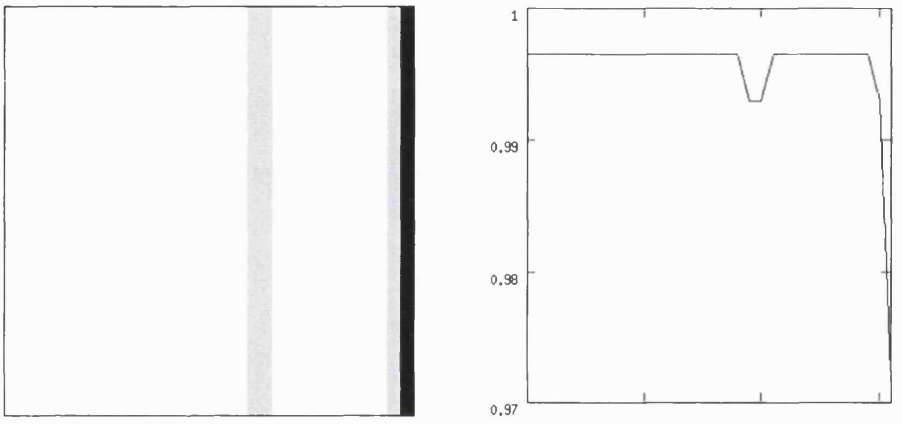
Figure 5-6, Segmentation of ramp

This, possibly unacceptable, result can easily be avoided by adjustment of the fitting parameters; raising σ and/or lowering α, β, or γ increases the connectivity values and, providing they all become positive, a one region segmentation results.

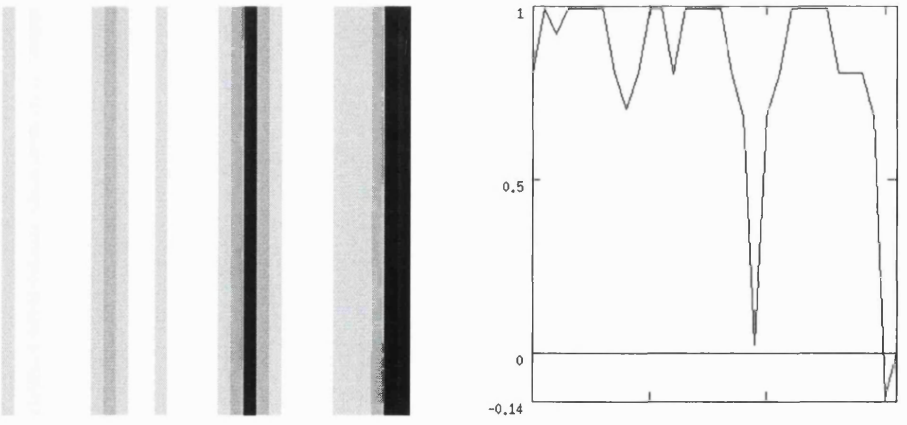
In Figure 5-7 the results for an originally perfect quadratic surface with a second order coefficient equal to 0.26 were shown. This produces a change of grey level from 0 to 249 over the 32^2 pixel image and is therefore quite a severe test. After integisation the one-byte image is no longer a perfect quadratic. The belief of fitting values are again all very high, the lowest occurring at the extreme right edge of the image. The differenced beliefs of connection/disconnection are again rather more variable with one negative value which would lead to a two region segmentation if these parameter values were used.



Grey levels



Beliefs of fitting



C_{pn} -D_{pn}

Figure 5-7, Quadratic results (c_{20}=0.26), σ=1.5, α=β=γ=1

5.6 Non-biquadratic surfaces

The following image, Figure 5-8, shows a simulated sphere positioned over a flat surface and illuminated from above. The shadow of the sphere can be seen on the flat surface. The light received from different points on the sphere's surface is dependent on the angles of incidence and reflectivity and on the scattering properties of the sphere's surface and hence is a complex but smoothly varying function.

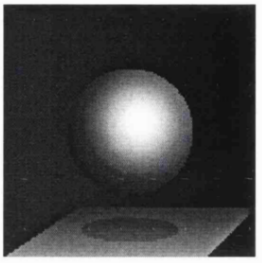


Figure 5-8 Ball image, 128^2 pixels

The belief of fitting (Figure 5-9) shows a barred structure in the background which is the outcome of this part of the image having a very gentle slope in grey level horizontally across the image and no variation in the vertical direction. Within the sphere the belief of fitting values show considerable variation but are all relatively high (greater than 0.76) with the greatest values being on those parts where the grey level variation is lowest and hence the fitting easiest. The only low (dark) values of the belief of fitting are in the corners of some of the regions and this is a result of the partially covered window effect which has already been discussed in section 5.4.

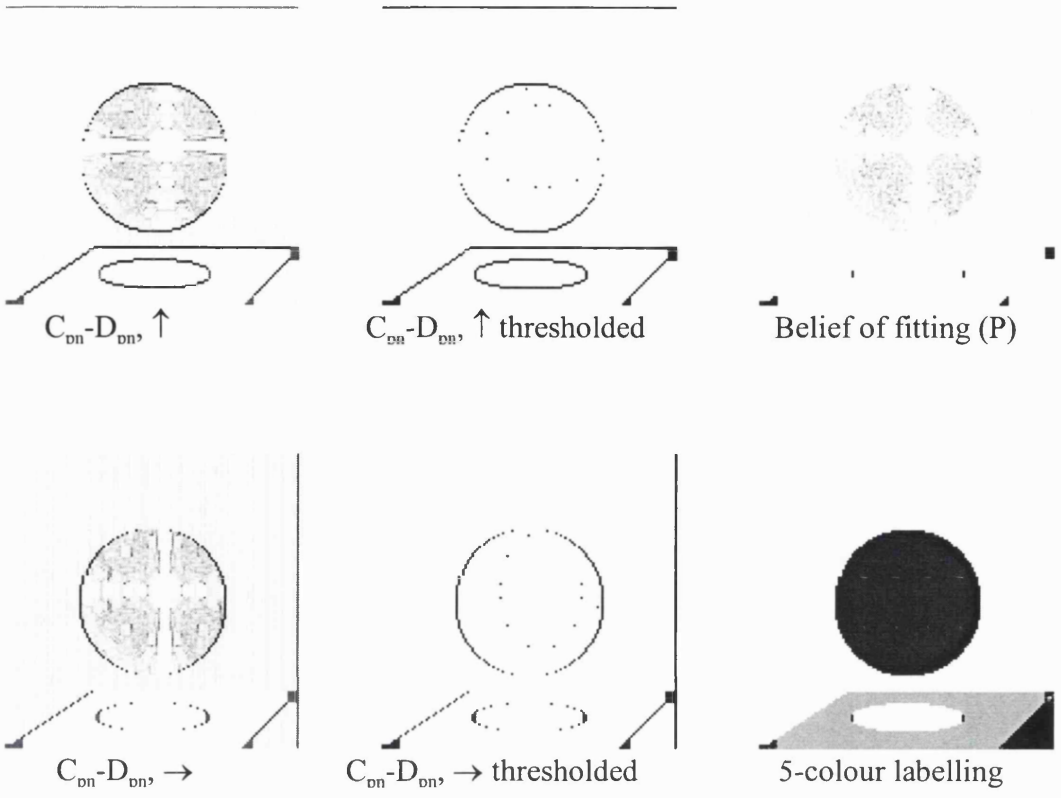


Figure 5-9 Ball image results, $\sigma=2$, $\alpha=\beta=\gamma=0.6$

The differenced beliefs of connection and disconnection, C_{pn}-D_{pn}, are shown in their original form and also thresholded at T = 0. These have high negative values within all surfaces showing strong connectivity and are only positive for pixel-neighbour pairs straddling boundaries. The five colour labelling shows that all the surfaces have been successfully delineated and the only defects in the segmentation are in the poorly fitted corners of some regions.

For completeness the biquadratic coefficients calculated for this image are shown; Figure 5-10.

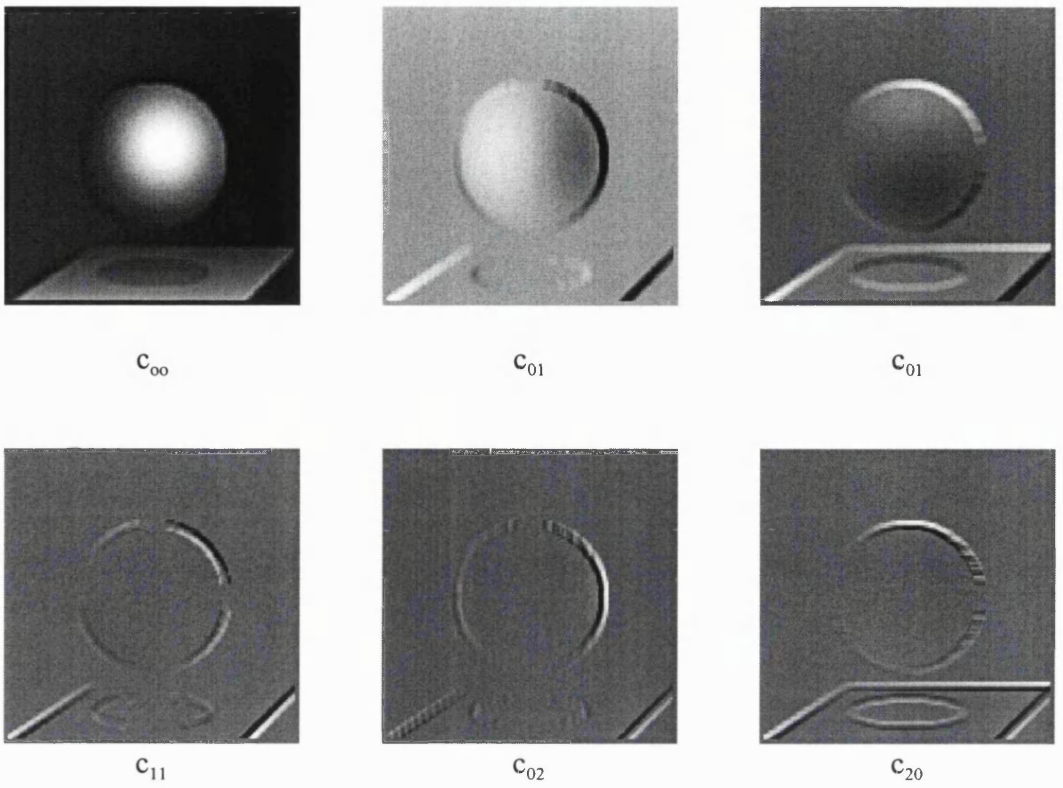
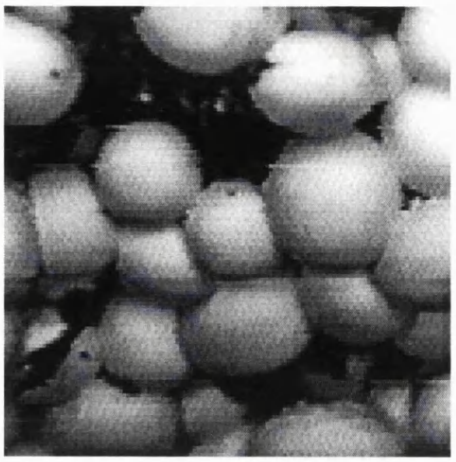


Figure 5-10 Ball image fitting coefficients

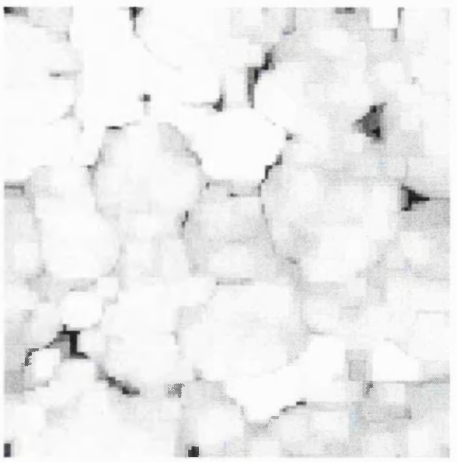
5.7 Real image

The image shown in Figure 5-11 is of growing mushrooms viewed from above and was supplied by The Silsoe Agricultural Research Institute. There is an interest in automatically finding the position and size of individual mushrooms to enable them to be picked by machine. The position of both the lighting source and imaging sensor relative to the mushrooms and the light reflective qualities of the mushrooms result in a variation of the grey levels imaged over the mushroom surfaces. The image is subject to the fault of 'tearing' which has the effect of a small displacement of some adjacent rows of pixels. In addition the general smooth grey level variation over the mushroom surfaces is affected, firstly by natural imperfections in the mushrooms themselves and also by small amounts of soil debris.

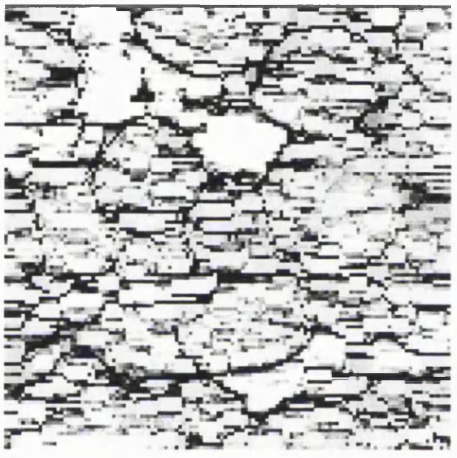
The belief-of-fitting image shows higher levels in the background regions than in the mushrooms themselves, presumably because the background, being almost constant in grey level with just a small amount of noise, is a relatively better fit than the grey level



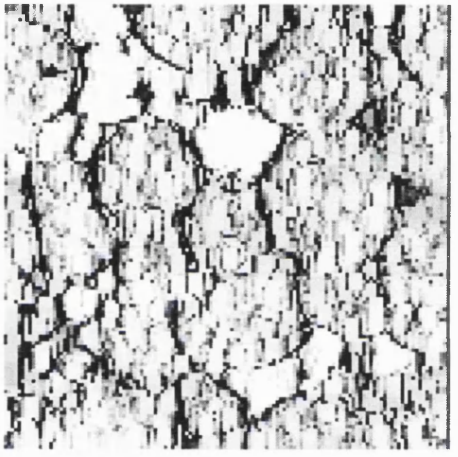
original image , 128^2 pixels



beliefs of fitting



C_{pn}-D_{pn}, ↑



C_{pn}-D_{pn}, →

Figure 5-11 Mushroom results, σ=6, α=β=0.8, γ=0.6

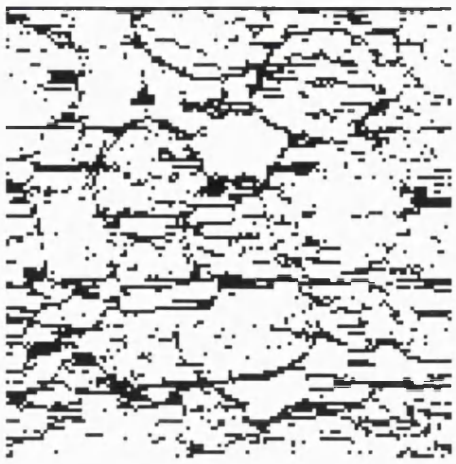
variation over the mushrooms. However the belief of fitting is in almost all parts a high level (>0.8). The exceptions are in regions too small to allow of a good fitting associated with small pieces of debris, small regions in the background and concavities at some boundaries. It is important to realise that for the best results the fitting beliefs should be everywhere high *including* at the edges.

The differenced beliefs of connection and disconnection show that the true mushroom boundaries have been fairly clearly extracted. The mushroom interiors have values generally showing very strong connectivity (~ +0.7) compared to the boundaries which have high negative values (~ -0.8). However not all parts of the boundaries are found; the large mushroom near the centre of the right half of the image is not totally separated from either of its two neighbouring mushrooms on the right.

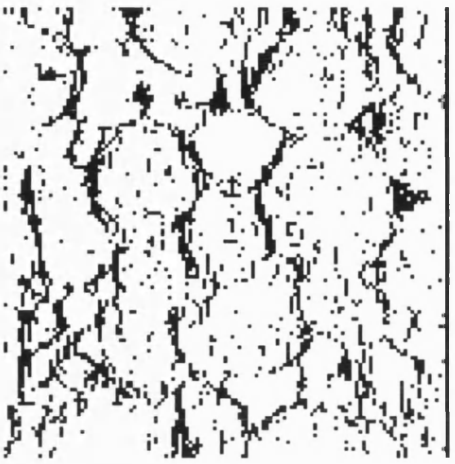
Figure 5-12 shows the differenced beliefs, in each direction, thresholded using T=0 and also the product of these results as this helps show the actual boundaries that have been found. Although all the boundaries have largely been found, close examination reveals that there are many small gaps and the result of this is that when the linked pixels are collected together into regions in the labelling process these incomplete boundaries are lost. This is exemplified in the region number labelling where many regions with incomplete boundaries have been lost. Due to the large number of regions produced (763) it is difficult to visually distinguish all of the regions especially those in the darker part of the image which have the lower region numbers.

In Figure 5-13 some individual regions extracted from this thresholded result are shown. This is achieved by the simple process of choosing a region label, setting all pixels with this value in the region labelled result equal to white (255) and all others to black (0). It can be seen that, whilst some individual mushrooms have been nicely delineated, albeit with the imperfections that can be expected from the real defects of the image, others have merged to form one composite region.

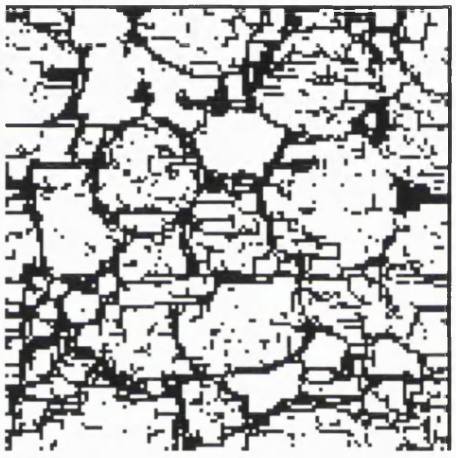
Figure 5-14 shows results from the same starting set of beliefs but using over-thresholding with T=0.2. The product of thresholded beliefs now shows a denser set of non-connected pixels and the region number labelling has resulted in more regions (1011) with an increased number of individual mushrooms being delineated although many are still merged with neighbouring regions.



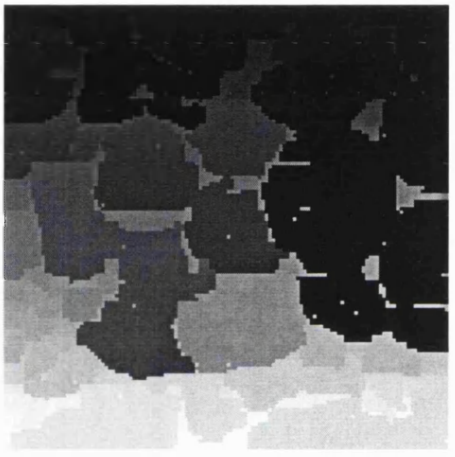
C_{pn}-D_{pn}, ↑



C_{pn}-D_{pn}, →



C_{pn}-D_{pn}, ↑*→



region number labelling

Figure 5-12, Mushroom, threshold T=0

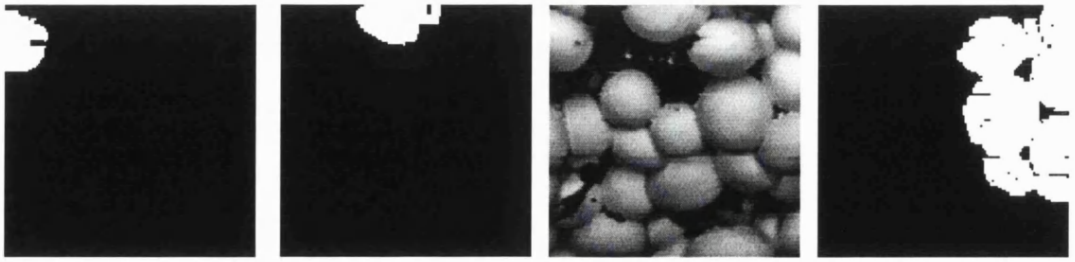
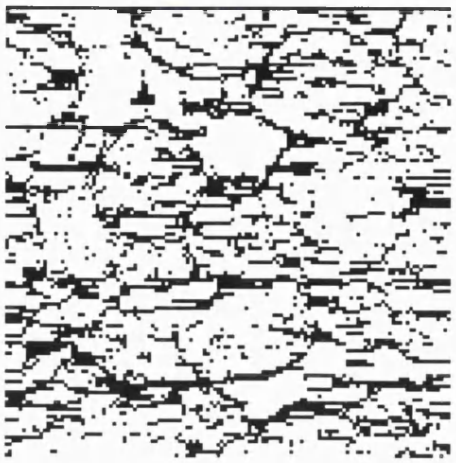
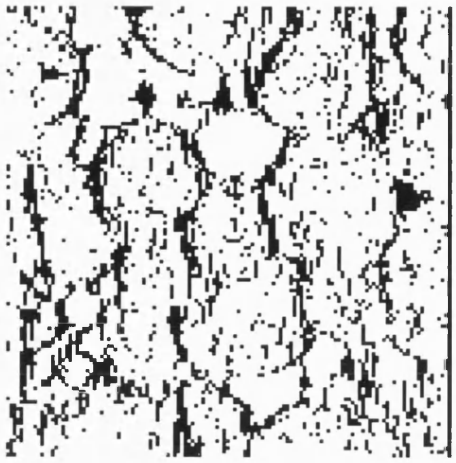


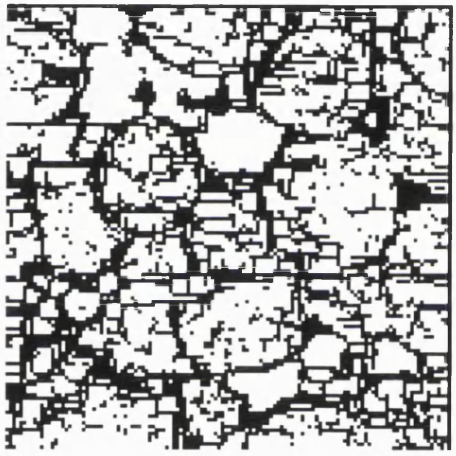
Figure 5-13, Mushroom regions from the T=0 segmentation



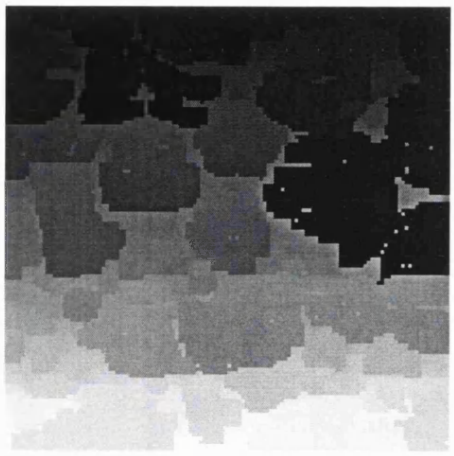
C_{pn}-D_{pn}, ↑,



C_{pn}-D_{pn}, →,



C_{pn}-D_{pn}, ↑*→,



region number labelling

Figure 5-14 Mushroom, threshold T = 0.2

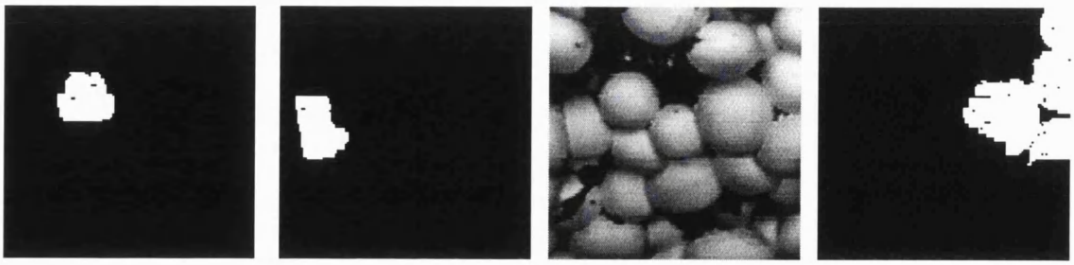
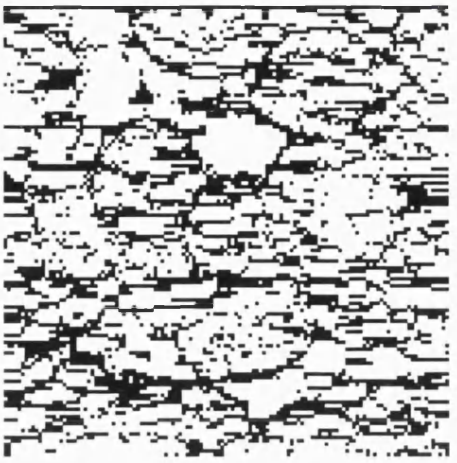
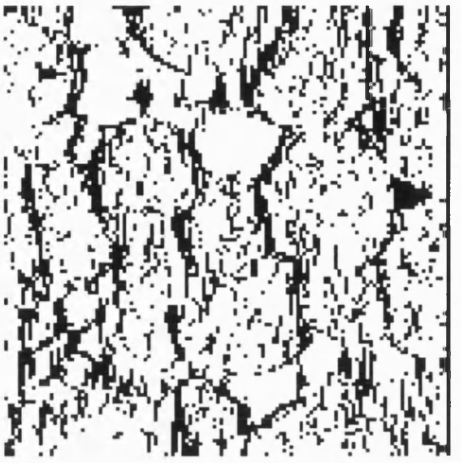


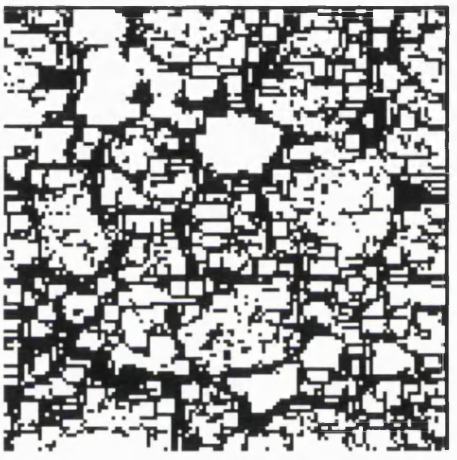
Figure 5-15 Mushroom regions from the T=0.2 segmentation



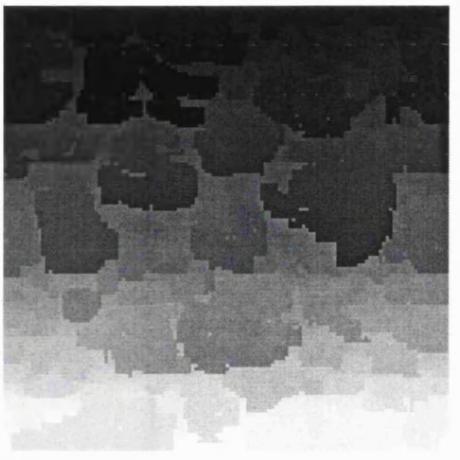
C_{pn}-D_{pn}, ↑,



C_{pn}-D_{pn}, ↑



C_{pn}-D_{pn}, ↑*→



region number labelling

Figure 5-16 Mushroom, threshold T = 0.4

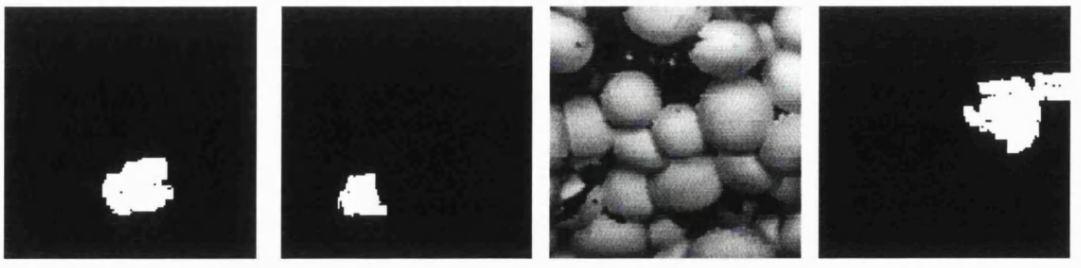


Figure 5-17 Mushroom regions from the T=0.4 segmentation

Increasing the threshold still further to T=0.4, produces the results given in Figure 5-16. There are now 1420 regions. The mushroom interiors are very dense with non-connected pixels. The labelled region result shows that a considerable proportion of the mushrooms have been individually extracted. However it is also now the case that some individual mushrooms have been split up into more than one region. It would be a relatively easy extension of the process to search the resulting segmentation for regions which are within a given size range, have a certain brightness and perhaps a certain convexity of shape and output their positions. Some, but not all, of the mushrooms would be found this way and this confirms the statements of Chapter 1 that the method being demonstrated is an image segmentation scheme and not a scene segmentation technique. No definition of the objects eventually required has been supplied to the method and therefore unless these objects happen to correspond to the image surfaces it should not be expected that they will all be found this way.

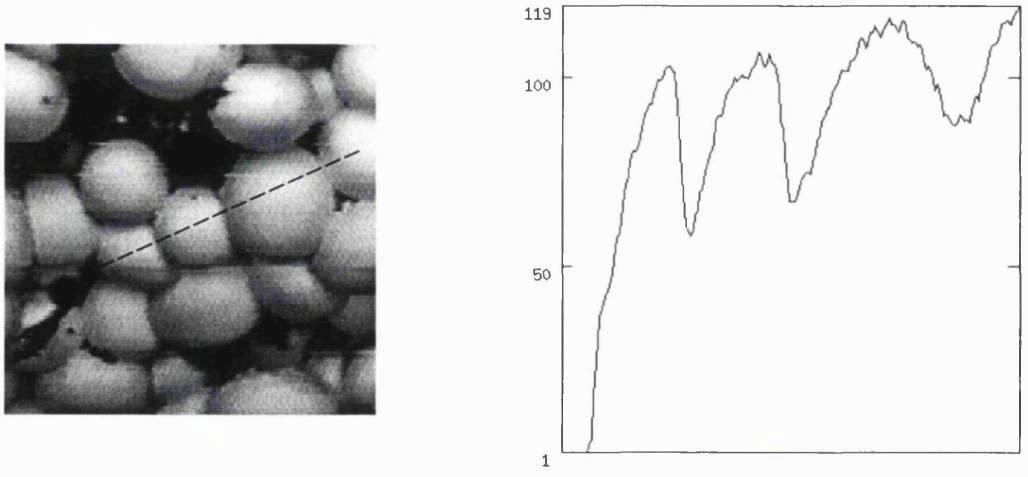


Figure 5-18, Mushroom grey levels along indicated line

Figure 5-18 shows a plot of the grey levels along a line selected to run through several adjoining mushrooms. This plot illustrates why it should not be expected that an image segmentation method will extract what are to the human eye and brain self-evidently separate objects. To the human eye-brain smooth continuous bright mushrooms separated by relatively dark edges seem evident but the plot shows that the changes in grey level at the mushroom junctions are not very different from those occurring at the mushroom centres. A method therefore which is based on finding grey level surface

changes will inevitably in this situation either lose certain mushroom boundaries in preserving the surface of individual mushrooms as one region or split up mushrooms into more than one region in attempting to find all the *true* boundaries. It is only by supplying contextual information, in this case about the imaging properties of the mushrooms, their shape and size, that objects rather than surfaces can be extracted.

5.8 Segmentation Comparison

It was beyond the scope of this work to produce a formal comparison with other methods. This was for two reasons; firstly this would have involved the implementation of many other techniques which was not feasible within the timescale of this research. Secondly a proper comparison would involve the definition of some measure of the quality of a given segmentation. This is a highly non-trivial task necessitating firstly the choice of a definitive set of test images, secondly a definition of their ideal segmentations and thirdly a quantitative measure for the comparison of two segmentations. All three parts are to some extent subjective and to add to this complexity is the fact that different segmentation techniques often demand different types and amounts prior information.

However a non-formal qualitative comparison with at least one other method is useful in illustrating relative differences and advantages.

A Markov random field (MRF) method was chosen because it is the basis of many current techniques and software was readily available for implementation. The MRF was briefly described in Chapter 3 and a more detailed description is given in Appendix B. The implementation here is a two-class system. The starting point for this is a manual segmentation dividing the image into two classes - object and background. The means and standard deviations of these classes are then measured and a threshold calculated at the crossing point of the two distributions. Using this threshold places each pixel in an initial class for which the class-conditional probability is a maximum. These classes are then optimised by minimisation of the energy function (see Appendix B). It should be noted that the classifications obtained here have not been converted into segmentations in which the regions would have different labels but this is a trivial process and here adds no extra visual information.

The ball image, Figure 5-19, used previously in this chapter and the segmentation already obtained using the surface fitting method are shown below.

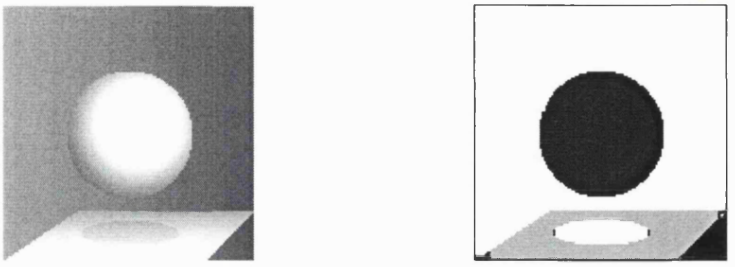


Figure 5-19, Ball image and segmentation using surface fitting

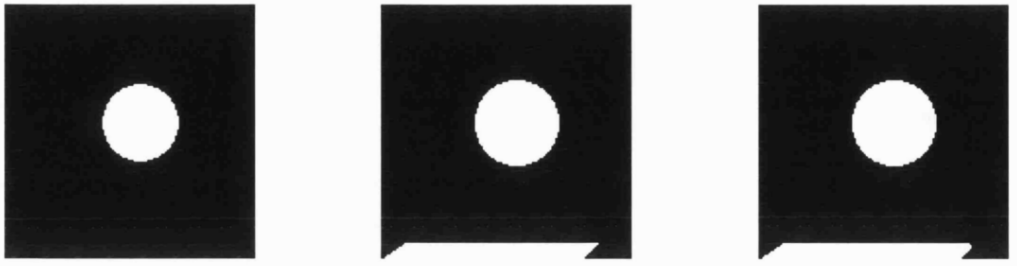
Figure 5-20 and Figure 5-21 show results obtained using the Markov Random Field method. These results use different manual starting segmentations; for example in Figure 5-20 the ball's exact outline is chosen. The grey levels used to calculate the statistics for class 1 - the ball - include both its bright and less bright areas and those for class 2 - the rest of the image - also include a mixture of pixels from the darker background and the brighter flat surface. This produced the statistics; $\mu_1 = 129.4$, $\sigma_1 = 64.5$, and $\mu_2 = 44.1$, $\sigma_2 = 39.1$ for the two classes, resulting in a discriminating threshold of 89.9 used for the initial classification.

manual segmentation initial classification classification after MRF

Figure 5-20, Markov results for ball image (1)

This initial classification places only the brighter parts of the ball and flat surface in class 1 and the rest of the pixels in class 2. Subsequent optimisation using the Markov method only slightly modifies the boundaries already achieved. In this work a value of 0.5 was used for the neighbourhood weighting parameter, θ (see Appendix B). Parts of the flat surface boundary do correspond to the true outline but the boundary of the ball itself has been lost and only the brighter part delineated.

In Figure 5-21 the brighter highlight of the ball was chosen to define class 1 statistics resulting in: $\mu_1 = 201.2$, $\sigma_1 = 29.2$, and $\mu_2 = 50.1$, $\sigma_2 = 41.8$ with a discriminating threshold of 136.2 used for the initial classification



manual segmentation initial classification classification after MRF

Figure 5-21, Markov results for ball image (2)

The higher threshold results in a smaller number of pixels placed in class 1 and hence in reduced sizes of both class 1 (white) regions. Again, the Markov optimisation produces little change and only part of the ball is outlined.

The Markov process was next applied to the mushroom image used previously (see section 5.7) Figure 5-22.

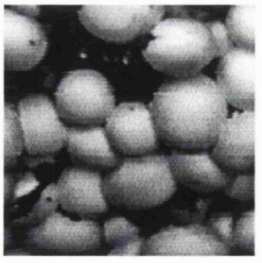
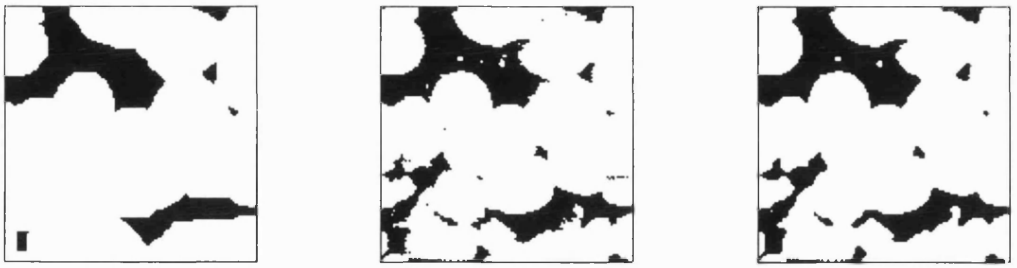


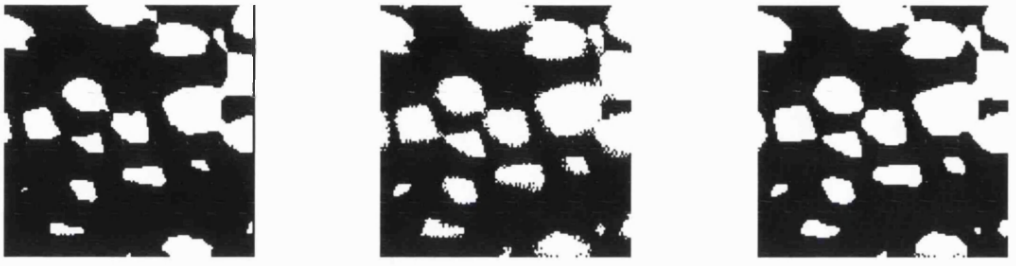
Figure 5-22, Mushroom image

Again two sets of results are presented based on the statistics for classes obtained from two different initial manual segmentations; Figure 5-23 and Figure 5-24.



manual segmentation initial classification classification after MRF

Figure 5-23, Markov results for mushroom image (1)



manual segmentation initial classification classification after MRF

Figure 5-24, Markov results for mushroom image (2)

For the mushroom image the initial classifications correspond more closely to the manual segmentations than was the case for the ball. This is a result of the rather more obvious two class nature of the image itself in which most parts of the mushrooms are brighter than most of the rest of the image. The Markov process again makes little change to the initial classifications merely smoothing the region boundaries and removing some of the very small regions. In neither example are the individual mushroom boundaries identified.

The method implemented is an optimisation procedure which alters the pixel class labels to minimise the Gibbs energy function. In theory, the starting labels used are unimportant as, providing the optimisation is applied correctly, the global minimum will be reached. In practice, however, it is far more efficient to progress from an already reasonable solution. The starting point chosen here is the maximum likelihood estimate which is essentially the minimum of the same energy function *ignoring* context; e.g. the dependence of the pixel classes on those of their neighbours. This is produced by application of the correct threshold.

The process is therefore in two stages; thresholding followed by an optimisation. The results of each have been portrayed separately and two general points can be made from the visual evidence presented:

- The initial thresholding produced very poor approximations to the genuine boundaries. This was true even when the class-conditional probabilities are optimally obtained (i.e; via manual delineation).

- The Markov based optimisation had very little effect on the final result

The explanation for these conclusions can be made as follows. When genuine adjacent regions have an overlap of grey levels then any threshold, however chosen, will inevitably lead to mislabelled pixels. This overlap can occur for three reasons. Firstly, noise present in adjacent regions will result in their histograms overlapping to a degree dependent on the noise variation (standard deviation) and the separation of the regions mean intensities. This effect is marginally present in the mushroom image where the low amount of noise produces small regions mislabelled in the mushrooms.

Secondly a smooth variation in the grey level of a region from above to below that of its neighbouring region will result in some overlap in their histograms. This is the effect seen in the ball image where the ball itself has a brightness varying from below that of the (nearly flat) background to considerably higher.

Lastly, when adjacent regions correspond to essentially the same class of object then the histograms are by definition the same and the overlap complete. This is the situation for neighbouring mushrooms in the image above - their histograms are virtually the same and hence inseparable by simple thresholding. The genuine boundaries are defined by changes in gradients and therefore any method not taking this into account will inevitably perform poorly. This situation also would clearly not be alleviated by allowing more than two classes; a constraint which might otherwise have been thought too restrictive.

The apparent ineffectuality of the Markov optimisation part of the procedure is explainable when the contextual dependence model is considered. This alters the anomalous labels wrongly produced by the initial threshold, and dictated by the grey level alone, to ones more in keeping with those of neighbouring pixels. This can work

well when the wrong labels are largely of single or small groups of pixels with a majority of correctly labelled neighbours. This is the case when the first type of histogram overlap, described above, caused by noise, is present. The few small regions incorrectly classed within the mushroom were subsequently relabelled. However when the incorrect labels are in large groups of pixels effectively reinforcing each other then the procedure can make little change. This was the situation with the ball image where a large part of the ball incorrectly labelled could not be restored by the Markov process.

An advantage of the method demonstrated and of thresholding in general is that it virtually guarantees the extraction of at least some parts of all regions. This contrasts with the surface fitting approach developed in this thesis in which regions can often be more accurately delineated at the risk of occasionally losing the ball or individual mushrooms. Techniques to overcome this apparent weakness will be developed in the following chapters of this thesis.

5.9 Summary

- A method of thresholding the beliefs of connection/disconnection to produce a set of links from which a segmentation can be formed was first defined.
- The zero-thresholded segmentation was shown to be a subset of the best segmentation.
- Over-thresholding was defined and the consequent effect of producing more non-linking and hence an increased number of regions described.
- The thresholding technique was applied to a variety of surface types starting with perfect biquadratics. Results with different parameter settings were analysed in some detail.
- The difficulty of surfaces with partially covered windows producing low fitting beliefs and hence undesirable non-linking and consequent multiple small regions was illustrated with a simple example.

- The effects of quantisation of images to a one-byte resolution was analysed and demonstrated with examples.

- The segmentation of a simulated image with non-biquadratic surfaces was described.

- A real mushroom image was then segmented using different thresholds and some (but not all) of the individual regions extracted shown to correspond to single mushrooms in the image.

- A comparison with another segmentation method was made.

- The work so far has demonstrated that some good results can be obtained by the simple technique of thresholding the connection beliefs. However it has been shown that over-thresholding, whilst necessary to ensure the completion of weak boundaries, does result in an over production of regions. The next chapters demonstrate a method of overcoming this problem.

6. Optimisation of segmentation

6.1 Necessity for optimisation of the over-thresholded segmentations

The previous chapter showed the use of the technique of thresholding the beliefs of connection and disconnection in order to obtain a set of links, $L(p,n)$, between pixels;

$$D_{pn} - C_{pn} < -T \Rightarrow L(p,n) = 1$$
$$D_{pn} - C_{pn} \geq -T \Rightarrow L(p,n) = 0 \qquad \text{Eq 6-42}$$

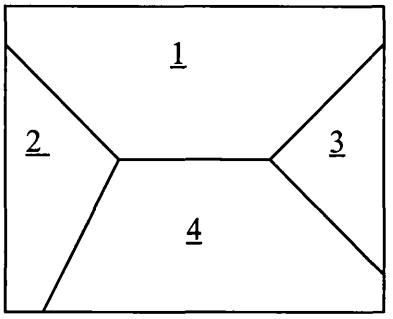
and hence a segmentation by accumulating linked pixels. If the threshold, T, is chosen to be zero then all the boundaries that are produced are part of the ideal 'best' segmentation and hence all of them must be kept. If the threshold is chosen to be greater than zero - over-thresholding - then the boundaries arising will include all of the above zero-thresholded subset of the ideal as well as, in general, some additional ones which, whilst they might not be exact components of the ideal segmentation, might nonetheless be an improvement in that the strength of segmentation is increased by their presence. However some of these additional boundaries will represent a devaluation of the segmentation and should be removed.

The problem is to decide which boundaries to keep in, and which to remove from, the over-thresholded results. Starting with the definition of the strength (see Chapter 4, Eq 4-20) of a particular segmentation consisting of R regions;

$$S = \sum_{r=1}^{R} \sum_{s=r+1}^{R} \Omega(r,s) \qquad \text{Eq 6-43}$$

in terms of the boundary strengths between neighbouring regions r and s. The over-thresholding means that some of the individual boundary strengths, $\Omega(r,s)$, might now be negative and at first it might be thought that deciding which to keep and which to lose is relatively easy; all that apparently needs to be done is to examine the strength of each boundary in turn and discard it if negative - a simple merging procedure. However the problem is more complex as the following simple example illustrates.

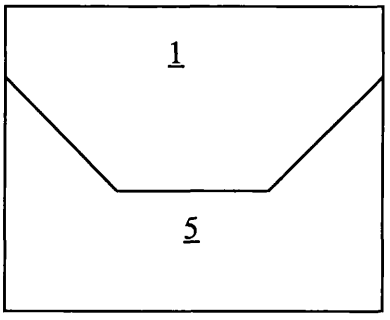
Consider a segmentation in which it is assumed that all of the $D_{pn} - C_{pn}$ values are greater than the reduced (negative) threshold which is used. The resulting four region segmentation is;



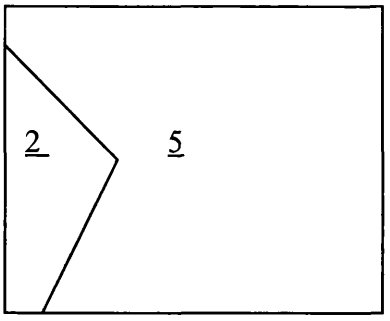
Further assume that the region boundary strengths are;

$$\Omega(1,2) = 10$$
$$\Omega(1,3) = 10$$
$$\Omega(2,4) = -4$$
$$\Omega(3,4) = -6$$
$$\Omega(1,4) = -8$$

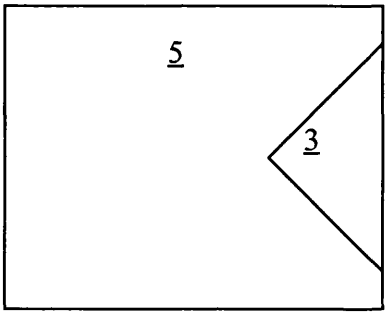
and therefore the strength of the four-region segmentation shown above is 2 (10+10-4-6-8). There are seven possible subsets of this simple segmentation each with its own segmentation strength. Only four have segmentation strengths which are non-negative. These are shown below with new regions given new region numbers.



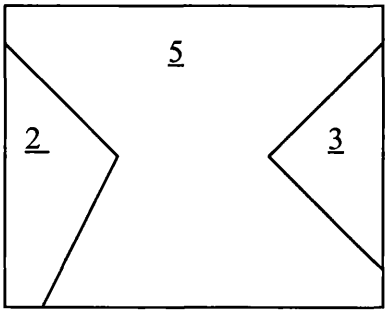
$$\begin{aligned}\Omega(1,5) &= \Omega(1,2) + \Omega(1,3) + \Omega(2,4) \\ &= 10 + 10 - 8 \\ &= 12\end{aligned}$$



$$\Omega(2,5) = \Omega(1,2) + \Omega(2,4)$$
$$= 10 - 4$$
$$= 6$$



$$\Omega(3,5) = \Omega(1,3) + \Omega(3,4)$$
$$= 10 - 6$$
$$= 4$$



$$\Omega(5,2) + \Omega(5,3) = \Omega(1,2) + \Omega(2,4)$$
$$+ \Omega(1,3) + \Omega(3,4)$$
$$= 10 - 4 + 10 - 6$$
$$= 10$$

It can be seen that the greatest segmentation strength is achieved by merging regions 2 and 3 with region 4 to obtain a two region segmentation.

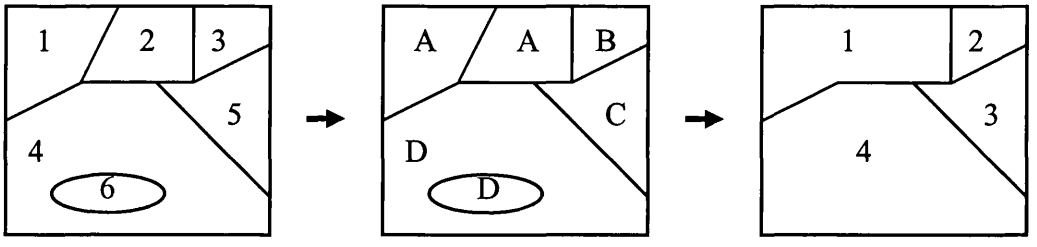
However, a merging procedure, examining boundaries in turn, might happen to investigate the boundary between regions 3 and 4 first and as (by definition) this has a negative boundary strength it would be lost, the regions merged and a weaker segmentation selected. Merging is a directional process and back-tracking is not possible; once two regions are merged they are lost, new boundaries formed and boundary strengths recalculated and if an incorrect decision is made at any stage then it is impossible to rectify later. Practical segmentations consist of many regions and a

merging procedure would be highly unlikely to produce the correct subset of boundaries.

Note that in the above example the weakest of the original boundaries is kept as part of the optimum segmentation. This shows that an algorithm which removed the boundary with the most negative strength, recalculated the new boundary strengths and then reiterated this procedure until no negative strength boundaries remained would not necessarily produce the optimum segmentation.

Examining all of the sub-segmentations and calculating their strength is impractical because of the number involved (see later for an estimate) and a different method of arriving at the optimum segmentation is needed.

Firstly a way of specifying the different sub-segmentations is needed. This can be done very conveniently by using a second set of labels and using these as the real designator as to whether or not neighbouring regions are merged. To avoid confusion the old region labels are termed 'region numbers' as these indeed will run from 1 to R, the number of regions. The new labels are called 'colours' as use is made of the 4-colour map theorem (see chapter 2 for a discussion) and only four used; A,B,C,D say. Given any new 'colour' labelling it is a relatively trivial formal process to produce a new 'region-number' labelling by simply accumulating connected groups of similarly coloured pixels and assigning the next available region number as illustrated in the following figure;



old region numbers colour labels new region numbers

Note that, although the colour labels are referenced to the old region numbers, they implicitly define a new set of region numbers. Defining any particular colour labelling

by L_a ,so that $L_a(r)$ = A or B or C or D for region numbers r = 1 to R, then the strength of the new sub-segmentation is given by a modification of Eq 6-43;

$$S_a = \sum_{\substack{r=1 \\ L_a(r)\neq L_a(s)}}^{R} \sum_{s=r+1}^{R} \Omega(r,s) \qquad \text{Eq 6-44}$$

which only takes account of the boundary between any pair of neighbouring regions if their colour labels are different. (Note that, in Eq 6-44, the subscript, (a), refers to the different possible *colour* labellings and not, as formerly in chapter 4, to the different region number segmentations, and therefore as the total number of regions, R, is here effectively fixed, it does not require the subscript (a)). The problem of deciding on the best sub-segmentation is then one of choosing the colour labelling, L_a, which maximises the total segmentation strength as given in Eq 6-44. It is emphasised again that this can only produce the best sub-segmentation from a given over-thresholded segmentation and this will not necessarily correspond to the best overall segmentation as defined originally in chapter 4.

Counting the number of possible sub-segmentations that are derivable from a starting N-region segmentation is equivalent to counting the number of segmentations that are possible for a given size of image which was outlined in chapter 2. Each of the N regions can be given any of the 4 colour labels giving a total of 4^N different colour combinations. However, each combination that uses all four colours appears in 4! permutations and therefore an estimate of the maximum number of possible sub-segmentations is $4^N/4!$. For even modest N this is an unfeasible number of segmentations to evaluate.

6.2 Overview of optimisation techniques

Firstly, the problem is slightly reformulated for later convenience. In the summation in Eq 6-44 note that Ω(r,s) is only non-zero for r and s which are neighbouring regions and hence it is only necessary to perform the summation over such neighbours. By defining the set of neighbouring regions of region r:

$\mathcal{N}_r \equiv$ {labels of regions neighbouring r}

and then defining the *Energy*, E_a, associated with a particular colour labelling, $L_a(r)$, of a given segmentation of a fixed set of regions, r = 1 to R, which is in effect the negative of the strength:

$$Energy = E_a = -\sum_{r=1}^{R} \sum_{\substack{s \in N_r \\ L_a(r) \neq L_a(s)}}^{R} \Omega(r,s) \qquad \text{Eq 6-45}$$

The *best* segmentation is now defined as that with the *lowest* energy. Note that each boundary strength is now effectively included twice but that this is of no significance as it does not affect the relative energies of any pair of segmentations which are all doubled as a result.

The total segmentation energy, E_a, can be written as a summation over individual region energies, $\varepsilon_a(r)$:

$$E_a = \sum_{r=1}^{R} \varepsilon_a(r)$$

$$\varepsilon_a(r) = -\sum_{\substack{s \in N_r \\ L_a(r) \neq L_a(s)}}^{R} \Omega(r,s) \qquad \text{Eq 6-46}$$

where each region's energy is simply the (negative) sum of the strengths of its boundaries with those of its neighbours which have a different colour label. As noted previously, each boundary strength, $\Omega(r,s)$, appears twice in the summation for the total energy; once when s is the neighbour of region r and once for the reverse situation ($\Omega(r,s) = \Omega(s,r)$) and, importantly for the optimisation procedure described below, that part of the total energy affected by the labelling of region r (say) is simply twice the region energy, $2*\varepsilon_a(r)$.

The problem of finding the optimum segmentation as now defined is one of a class of global optimisation problems which has become of increasing importance in the past few decades. Horst and Pardalos[84] review many of the techniques of global optimisation

Global optimisation is the task of finding the best set of parameters to optimise an objective function. In general, there are solutions that are locally but not globally

optimal and consequently, global optimisation problems are typically quite difficult to solve exactly. In the context of combinatorial problems, they are often of the non-deterministic polynomial type; the number of solutions grows as an exponential function of the problem size - *not* as a simple polynomial function of the size. Global optimisation problems fall within the broader class of nonlinear programming (NLP). A NLP problem has the general form:

Find a vector of reals, x, to minimise F(x)

subject to the constraints: $g_i(x) = 0$ for $i=1,...,m_1$ where $m_1 >= 0$

and $h_j(x) >= 0$ for $j=m_1+1,...,m$ where $m >= m_1$

Some of the more important classes of global optimisation problems, as listed by Pinter[85], are differential convex optimisation, complementary problems, minimax problems, bilinear and biconvex programming, continuous global optimisation and quadratic programming.

Combinatorial problems have a linear or nonlinear function defined over a set of solutions that is finite but very large. Typically there is a finite collection of subsets; $\{S_1, S_2, ..., S_m\}$ say. For each subset, S_k, there is an objective function value, $f(S_k)$, and the problem is to optimise $f(S_k)$; that is to find the optimum subset. The feasible subsets are represented by the inclusion or exclusion of more basic fixed value elements such that they satisfy certain conditions. This is a special class of *integer programs* whose decision variables are binary valued: x(i,k)=1 if the i-th element is in S_k; otherwise, x(i,k)=0. (formulation of an *integer program* is not always easy, and often there is more than one formulation, some better than others.)

Examples of combinatorial optimisation problems are; assignment, networks, covering, matching, packing, routing, scheduling, shortest path and 'travelling salesman'. Heuristic methods like simulated annealing, tabu search and genetic algorithms have been successfully applied to these problems to find approximate solutions.

The segmentation optimisation problem defined here is one of this combinatorial type: each segmentation subset is one possible colour labelling $L_k(r)$, and the elements are the boundary strengths, $\Omega(r,s)$. The inclusion, or not, of the element $\Omega(r,s)$ in the subset S_k is given by $x((r,s), k) = 1$ if $L_k(r) = L_k(s)$.

Here simulated annealing has been chosen as an optimisation method as it has achieved the reputation of an extremely powerful method which guarantees success (if correctly applied) and it is relatively simple to implement.

6.3 Simulated annealing as applied to regions

Simulated annealing is a technique that is especially suitable for optimisation problems of large scale where the desired global extremum is hidden among many poorer local extrema. The ability to find the global optimum is not related to the initial conditions (i.e., the starting point).

The basis of the method is an analogy with the thermodynamics of metals cooled slowly, *annealed*, from a high temperature to form very large crystals or, more generally, of liquids freezing and crystallising. At high temperatures the molecules are able to move freely but as the temperature is lowered this mobility is reduced. The molecules line up into the crystal state appropriate to their chemical properties and, provided the cooling is sufficiently slow, the crystal sizes can be correspondingly large. A single crystal is the minimum energy state. If the metal is cooled quickly, *quenched*, a polycrystalline state is reached instead, which has a higher energy.

The Boltzman probability distribution,

$$\Pr(E) \approx \exp(-E/kT)$$

expresses the idea that a system in thermal equilibrium at temperature T has its energy distributed among all different energy states E. The parameter, k, is the Boltzman constant. Even at a low temperature there is still a small chance of the system being in or moving to a high energy state. Correspondingly there is a chance for a system to move from a local energy minimum state to a higher energy state and thence to a better, more global, energy minimum.

These ideas were first used to enable optimisation in numerical problems by Metropolis et al[86]. A system is assumed to change its configuration from one with energy E_1 to one with energy E_2 with probability, $p = \exp[-(E_2 - E_1)/T]$. Here the Boltzman constant is rendered redundant as now both the energy and temperature are only analogies and can be expressed in arbitrary units. If E_2 is less than E_1 then p is

greater than unity and in this case p is set to one: the system is bound to assume this change.

An algorithm to apply simulated annealing to the colour labels of a fixed set of regions based on the supplied region boundary strengths is formulated as follows:

- Choose a random region number; r
- Calculate all possible energy contributions for this region,

$$\varepsilon(r) = -\sum_{\substack{s \in N_r \\ L_a(r) \neq L_a(s)}}^{R_a} \Omega(r,s),$$

  varying its labelling over all four possible colours; $L_a(r)$ =0,1,2,3,
- Store the energy, $\varepsilon_{cur}(r)$, associated with the *current* labelling, $L_a^{cur}(r)$.
- Calculate the minimum energy of the other three labellings - the possible *new* energy - $\varepsilon_{new}(r)$ and store it and the associated labelling $L_a^{new}(r)$
- If $\varepsilon_{new}(r) < \varepsilon_{cur}(r)$ then accept the new label; relabel to $L_a^{new}(r)$.

  else

  if $\varepsilon_{new}(r) \geq \varepsilon_{cur}(r)$, then accept the new label with probability

  $p = \exp[-(\varepsilon_{new}(r)-\varepsilon_{cur}(r))/T]$

  [this is done by first calculating the probability, p, as above and then generating a random number, q, in the range (0,1). If $q < p$ then the new label is accepted]
- Repeat the above steps a large number of times relative to the number of regions
- Reduce the temperature according to the annealing schedule and repeat the above steps

This method was first applied to a test image as shown in Figure 6-1. This consists of an internal series of alternating bright and dark bands surrounded by two other regions, each of which is in contact with all the internal regions as well as with each other. Each

of the regions is simply a constant grey level and hence segmentation is trivial as the region boundaries are well defined. The region numbers are also shown in Figure 6-1.

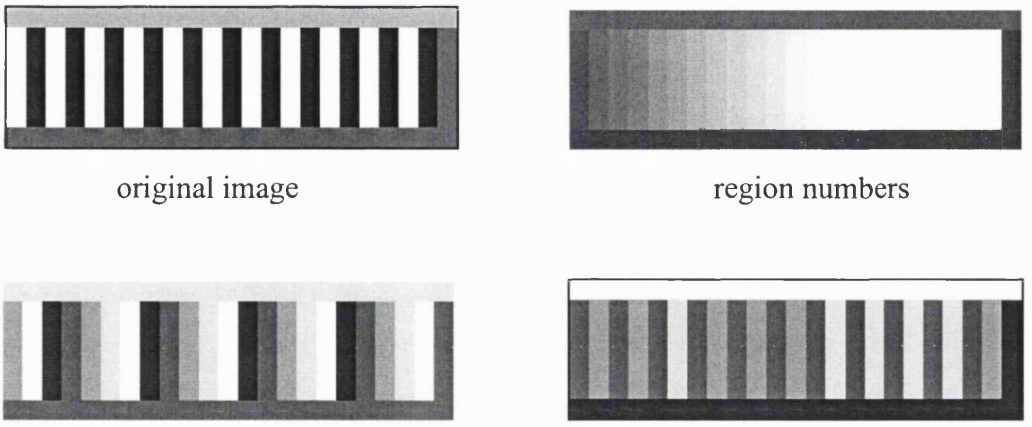


Figure 6-1 Test image for four colour labelling

Now, if this set of regions were to be correctly colour labelled using the theoretical minimum number of four colours, then, the only way this could be achieved would be by using two labels for the internal regions, alternating them region by region and the remaining two labels for the outer pair of regions (which, visually, would look very similar to the original image as only two different grey levels were used for the inner set of regions).

Attempts to achieve a correct colour labelling using the above simulated annealing algorithm with four allowed colours invariably failed; the resultant colour labellings always used three colours for the inner set of regions and the same colour for the outer pair. This is indeed very close to an ideal labelling as the only imperfection is in the small boundary of the two outer regions; the inner regions would all be correctly labelled differently from their neighbours. The difficulty clearly lay in making a major change of labelling from three to two labels for the inner set for what is only a relatively tiny reduction in the segmentation energy.

The fundamental cause of the problem was attempting to use the theoretical minimum of four colours which was too restrictive and this was avoided in subsequent work by changing to using five colours. Figure 6-1 shows five colours used to initialise the annealing procedure which here was done by simply assigning the colours 0, 1, 2, 3, 4,

in turn, to the region numbers resulting in the ordered pattern. (In subsequent work a random assignment of the initial coloration was used).

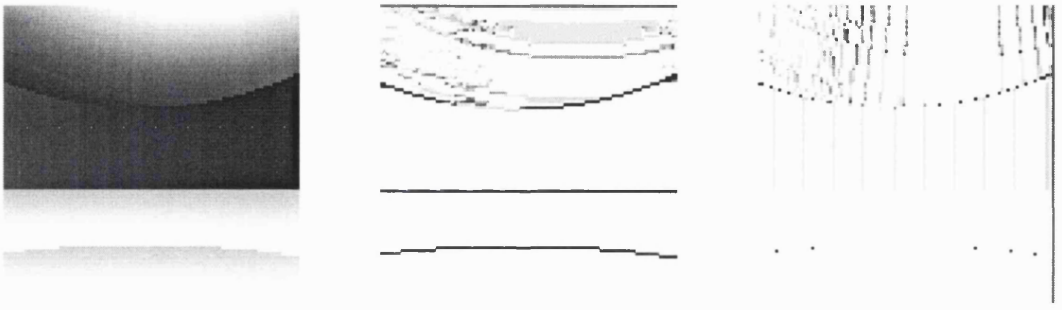
The results of the annealing procedure are also shown and here the outer two regions have been assigned colours 0 and 1 and the inner three regions share the remaining colours 1, 2, and 3. This result happened to use a starting temperature of 8 and reduced this in 16 steps by 0.5 and considered 1600 possible region colour changes at each step. Further considerations of the theory behind the choice of annealing schedule will be given later: the important point here is that the result was consistently stable and perfect over a considerable range of schedules.

This change (from four to five colours) has no consequences for the definitions of best segmentation; any number of colour labels (>3) can be used in Eq 6-45 without affecting its validity. In practical terms five colours have considerably more combinations of correct labellings than four and the number of different minima that correspond to a best segmentation are increased but this does not appear to have caused any real problems in subsequent work.

It might be argued that the image used to demonstrate this problem is somewhat artificial and that such combinations of regions are unlikely in real images and further that the segmentation produced with four colours is substantially correct so perhaps the change to five colours is unnecessary. Conversely, however, perhaps it would be possible to construct an (even more unlikely) image which proved difficult to colour correctly with just five colours. Overall the use of this inconsequential change seemed felt justified.

The image shown in Figure 6-2 is part of the simulated ball image used in Chapter 4 (fig 4-8). The previous work used this image at a reduced resolution to produce very good results with simple thresholding. At full resolution this method does not work so satisfactorily. The belief of fitting values are everywhere very high and are not shown here. The connection/disconnection belief differences show that the lower two surfaces are strongly connected internally and well-separated from their neighbours. The middle surface (below the ball) is a low gradient ramp in grey level horizontally and as a consequence shows vertical bands of reduced connectivity. The ball's more complex grey level structure has produced a considerably complicated connectivity pattern

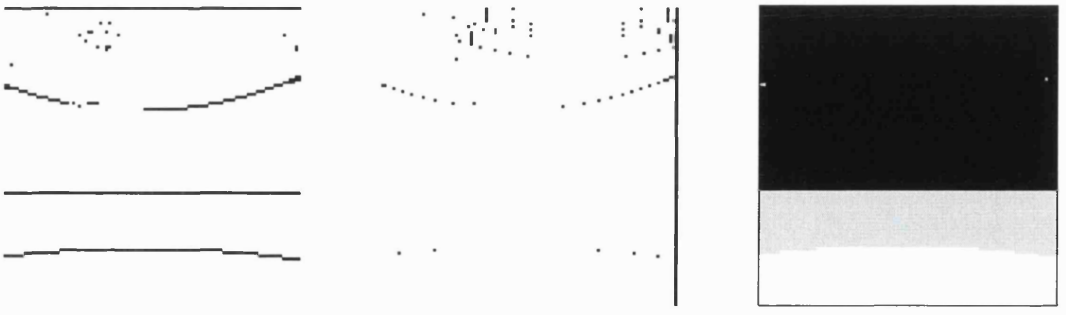
internally although all the values are positive (indicating connectedness rather than disconnectedness) but, more importantly, the generally strong (~ -1) boundary between the ball and lower region is very weak at the middle point.



'ball' image 100^2 pixels C_{pn}-D_{pn} ↑ C_{pn}-D_{pn} →

Figure 6-2 Ball belief values; σ =3, $\alpha = \beta = 1.2$, $\gamma = 1$

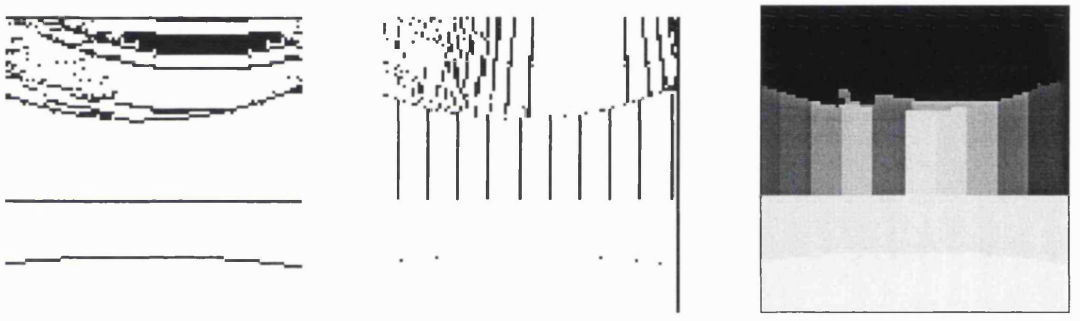
Thresholding the beliefs at zero produces the results shown in Figure 6-3 (positive = white = connected). The internal surfaces are seen to be largely clear of disconnected values but the ball boundary is distinctly broken. Segmentation using these values produces the image shown with the ball and the adjacent lower region merged together - a clearly unsatisfactory result.



C_{pn}-D_{pn} ↑ C_{pn}-D_{pn} → regions

Figure 6-3 Ball results thresholded at 0

Choosing a higher threshold produces a considerably denser set of disconnected belief values and the resulting segmentation, Figure 6-4, has many more regions (97). In particular the middle region is now split into vertical bands corresponding to the ramp steps. The ball itself is divided into a number of variously sized regions but, importantly, is now everywhere separate from its *true* neighbouring regions. Comparing this segmentation with the original belief values, Figure 6-2, it is evident that the undesirable region boundaries, not corresponding to true surface junctions, must have only very weak disconnectivity values and hence the segmentation is a good subject for optimisation.



C_{pn}-D_{pn} ↑ C_{pn}-D_{pn} → regions (97)

Figure 6-4 Ball results thresholded at 0.8

Annealing of the region set depicted in Figure 6-4 is shown in Figure 6-5. The starting temperature was chosen from trials as 7 (see later for work on the choice of an annealing schedule). Figure 6-5 shows the number of region label changes *considered,* many of which would be rejected through not passing the probability test for acceptance. The first 800 changes considered are almost sufficient to complete the correct labelling of the easier surfaces for which the connection beliefs are relatively high although the labelling of the lower two regions does change in the next 800 changes at a reduced temperature. The remaining regions within the ball interior are gradually dealt with as a lower temperature is proceeded to.

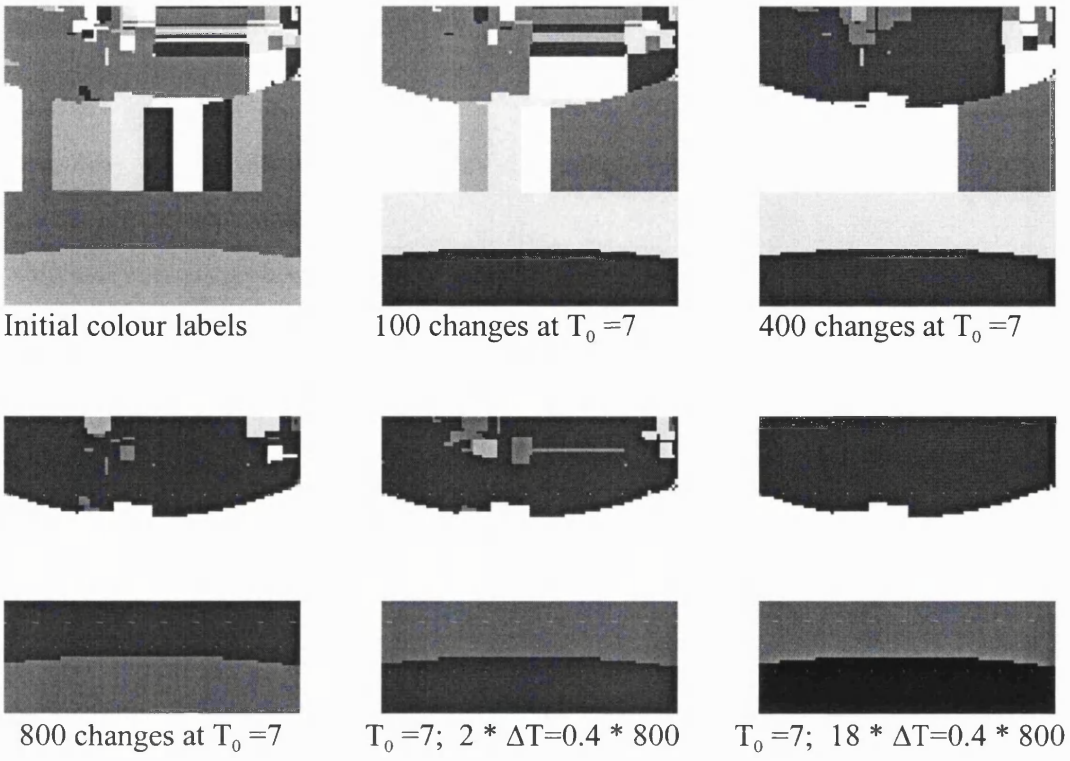


Figure 6-5 Ball image simulated annealing results

All the major surfaces, including the ball itself, are correctly delineated with a single colour label and this is clearly a major improvement on the results using zero-thresholded non-annealed results (Figure 6-3).

In Figure 6-6 are shown the results of using part of an image of mushrooms similar to that used previously (see chapter 5). The fitting beliefs are generally high everywhere except in areas not fully covered by the 5 by 5 window used - for example some parts of the narrow gaps between the mushrooms. The connection beliefs (only the vertical components are shown for brevity) show good separation of the different surfaces. When zero-thresholded however the resulting segmentation has only 50 regions and many of the evident surfaces are merged together.

Over-thresholding produces many more regions (524) with surfaces clearly split up. Application of the simulated annealing process produces a visibly good result and renumbering of the optimised colour labels shows that the total number of regions is now reduced to 107. Of these many (although not all) are individual mushrooms which can be simply picked out.

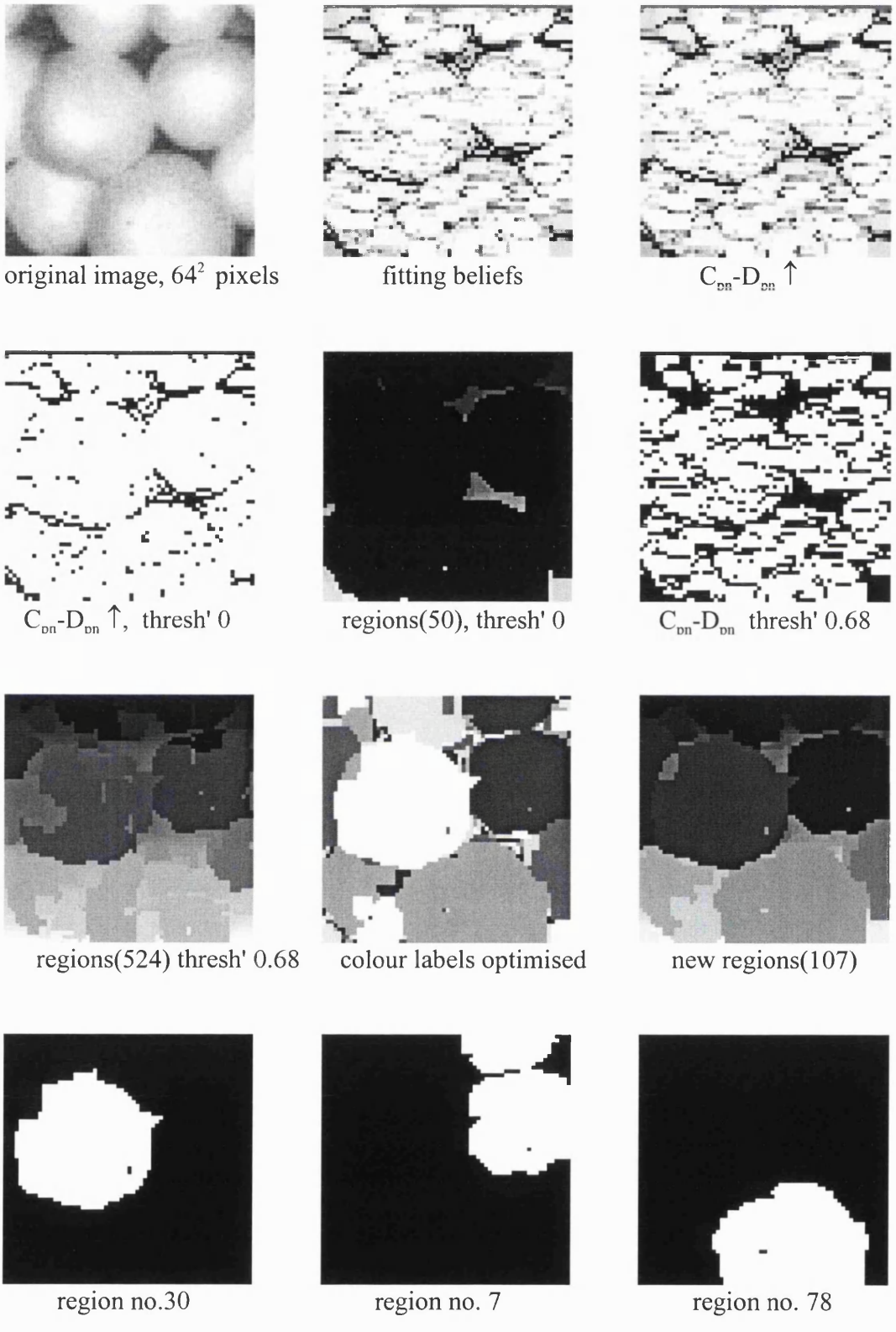


Figure 6-6 Mushroom image, over-thresholded result optimised, renumbered. σ=8, α=β=0.5, γ=0.2. Optimised results using T_0=5, 21 steps of 0.25 (to zero), 50000 labelling changes considered at each step

6.4 Summary

- First it was shown that a simple merging of the over-thresholded segmentation results produced in Chapter 5 would not necessarily be satisfactory and that a more sophisticated optimisation procedure is desirable.
- The use of four-colour labelling to record the results of the optimisation was demonstrated.
- A review of optimisation techniques was made and the problem put into a suitable format.
- The details of the method of simulated annealing as applied to the over-thresholded regions were explained.
- The reasons for a change to five (from four) colours were given.
- The method was demonstrated successfully on the 'ball' image and on a 'mushroom' image.

7. Optimisation of segmentation at different scales

7.1 Simulated Annealing applied to regions obtained within sub-images

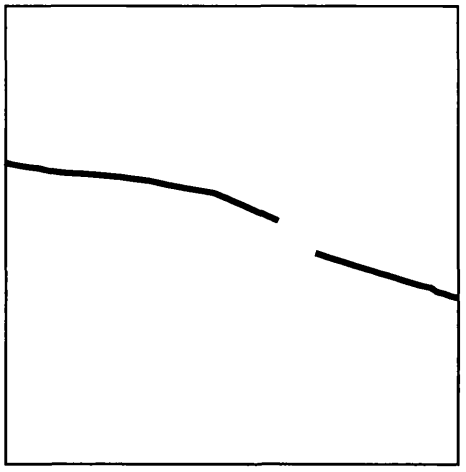
A problem depicted in part by the work of the previous section and referred to elsewhere in this thesis is that of the loss of regions caused by an incomplete boundary. The previous section showed one way of dealing with this *if* the boundary could be completed by over-thresholding at the expense of introducing some unwanted regions which were then effectively removed by annealing. However it is possible for parts of some boundaries to be so weak or simply absent that no amount of over-thresholding can be effective. This section shows one method of overcoming this problem. The basis of the method is the idea that, although a boundary might be incomplete over the whole image, it is bound to be complete over certain smaller parts of the image - sub-images.

The first part of the segmentation process described so far - computation of the beliefs of connection/disconnection followed by thresholding to produce linked/unlinked pixel-neighbour pairs - is completed as before. But then the image is divided into a grid of adjoining non-overlapping sub-images and it is within each sub-image that linked pixels are grouped into regions none of which can be larger than the chosen sub-image size. The resulting overall segmentation will contain, as a result of the method, all the sub-image edges and often these will be false boundaries separating regions which are in fact parts of the same surface. Additionally, however, providing the sub-image size is chosen appropriately, substantial parts of previously completely lost boundaries will now be retained.

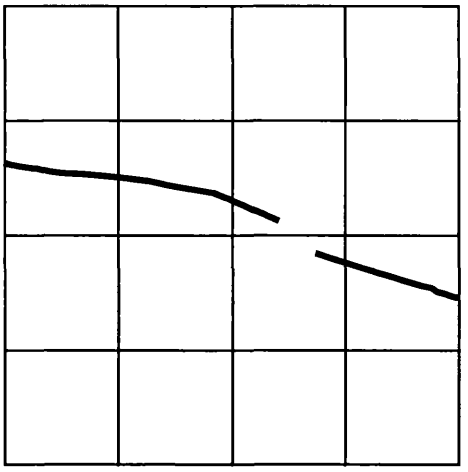
Optimisation of this segmentation will result in the removal of some of the false forced sub-image boundaries whilst keeping parts of real incomplete boundaries which have been found, to produce the best sub-segmentation. The previously incomplete boundaries will be completed by retention of parts of some of the sub-image edges producing an optimised segmentation which, whilst it will not perhaps exactly coincide with the ideal segmentation, will be a better approximation than results obtained from the method applied to the image as a whole.

This process is illustrated schematically in Figure 7-1. A few explanatory points are worth making. Firstly, purely for brevity, an artificially simple image of the differences of beliefs of connection/disconnection is shown in which white represents absolute connection and black absolute disconnection. Division is into only 16 sub-images: in practice the sub-image size relative to the original image might well be much smaller. Region numbering within each sub-image starts from where the last one left off to ensure that region numbers taken over the whole image are unique.

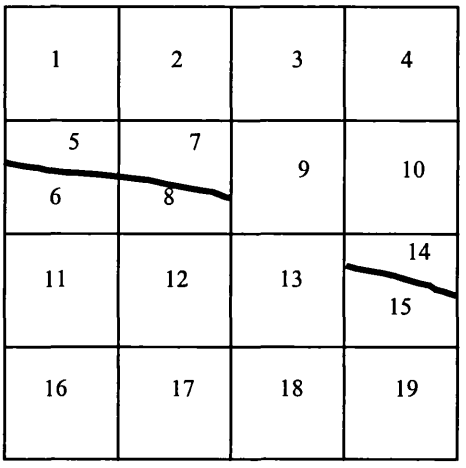
Optimisation of the segmentation is performed using colour labelling as described in the previous section. In this especially simple image only two colours are necessary. The final segmentation into two regions is to be compared with the single region result which would be obtained by applying the method to the image as a whole (irrespective of the threshold on beliefs used). The somewhat disjointed region boundary could be made to correspond closer and closer to the ideal smooth boundary by choosing smaller and smaller sub-image sizes - this will be discussed further later in this chapter.



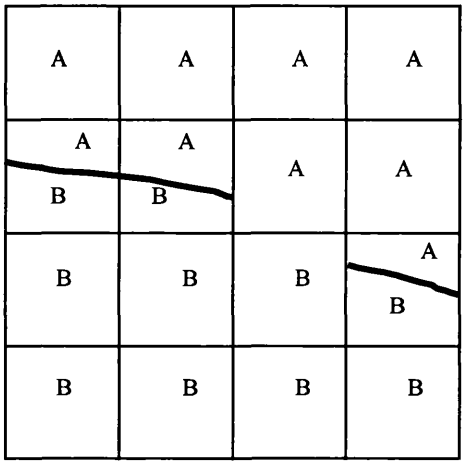
Beliefs , C_{pn} - D_{pn}



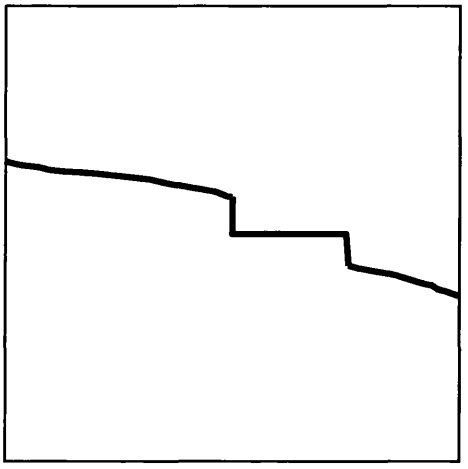
Division into sub-images



Segmentation in sub-images



Optimisation using colour labelling



Final segmentation

Figure 7-1 Schematic representation of segmentation within sub-images followed by annealing of the regions

Before applying this method to more complex images it is instructive to analyse the annealing of the simplest possible image in which all the grey levels have the same value. For this image the differences of beliefs of disconnection and connection (D_{pn}-C_{pn}) will be everywhere -1 indicating absolute connectivity and this will clearly still be the case after thresholding. Consider a 100 by 100 pixel image and use a sub-image size of 10 by 10 producing 100 sub-images each of which forms, after grouping its connected pixels, a single region; see Figure 7-2. (The original image and the differenced beliefs are not shown here because, consisting as they do of exactly the same pixel values over the entire image, they are somewhat uninteresting to view.)

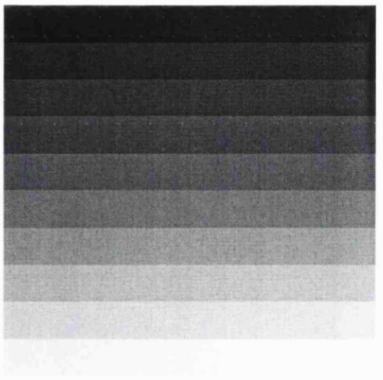


Figure 7-2 Region-number segmentation of 100 (10 by 10 pixel) sub-images

After annealing, the expected result is for all the sub-image regions to carry the same label forming the perfect (for this image) segmentation of one region. Now each of these sub-image regions forms a boundary with up to four neighbouring regions. Each boundary consists of 10 pixel-neighbour pairs and each pair has a strength, as previously stated, given by D_{pn}-C_{pn} = -1. The total boundary strength is therefore given by $\Omega(r,s)$ = -10, a negative strength indicating that the boundary is in fact false. Assume that the simulated annealing process is being applied *at a constant temperature T*; that the colour labels of all the regions have reached the same value (the correct solution), and that one of the internal regions (with four neighbours) is chosen next. The current region energy, with all the colour labels being the same and no boundary strengths included is zero (see previous chapter, Eq 6-5). Changing the region's label to any other will cause all four neighbour boundaries to be included giving a new region energy of +40. As this energy change is positive it is only instigated with a probability, exp(-40/T) As there are 64 such internal sub-image

regions it would expected that, if the annealing were stopped at any stage, then the number of internal regions mislabelled as different from their neighbours will be, on average, 64*exp(-40/T). In addition to the internal regions there are 32 sub-image regions on the rim of the image each with three neighbours and four corner regions each with only two neighbours. Similar reasoning shows that it would be expected for these to produce 32*exp(-30/T) and 4*exp(-20/T) mislabelled regions respectively.

Table 7-1 gives some results predicted by these formulae and also the actual numbers of regions mislabelled when the optimisation process, *at a constant temperature*, had reached a steady state which was judged to be the case when the bulk of the colour labels had stabilised to the same value and the number of differently labelled regions was fairly steady (although the actual regions so labelled and the labels used were changing).

Temp, T	Average number of regions mislabelled					No.of label changes to steady state
	Internal, 64exp(-40/T)	Edges, 32exp(-30/T)	Corners, 4exp(-20/T)	Total predicted	Actual number	
3	0.00	0.00	0.01	0.01	0.0	38000
5	0.02	0.08	0.07	0.17	0.0	14800
6	0.08	0.22	0.14	0.44	0.7	8900
7	0.21	0.44	0.23	0.88	2.1	8800
8	0.43	0.75	0.33	1.51	4.5	6600
9	0.75	1.14	0.43	2.32	7.6	7300
10	1.17	1.59	0.54	3.30	11.3	10000
12	2.28	2.63	0.76	5.67	27.4	10800

Table 7-1 Numbers of 100 sub-image regions mislabelled using simulated annealing

The actual numbers in Table 7-1 were taken as the average of a small number (~8) of runs, enough to give a sufficiently good approximation. It can be seen that, at the lower temperatures (T = 3 $\rightarrow$ 6), the predicted and actual numbers are in very close agreement but that as the temperature is raised the actual numbers begin to exceed

those predicted by a significant amount. The explanation for this difference lies in the relatively small total number of sub-image regions considered (100). The formulae assume that there are 64 internal regions with four neighbours, 32 edge regions with three neighbours and 4 corner regions with 2 neighbours. But if, for example, an edge region is mislabelled then two of its previously edge-type neighbours effectively become corner-types with an energy change of only +20 (rather than +30) now required to mislabel them. Similarly, the other, formerly internal-type, region becomes an edge-type. So, as more regions are mislabelled, the more likely it is that other neighbouring regions are also mislabelled to the same value and blocks of such regions will grow. An avalanche type effect occurs which means that, at a high enough temperature, there is no single colour label over most of the image but rather all five labels are utilised in an irregular blocky structure. This effect can be seen in Table 7-1 as the temperature is increased from T=7.

The last column of Table 7-1 shows the number of region label changes considered before the steady state condition was achieved. These figures are only rough estimates of what is a subjective end-state (although it would be perfectly possible to define formally using the changing statistics of the numbers of different labels used). The figures show that at low temperatures the end state requires many region label changes to be considered before it is achieved. This is to be expected as, at a lower temperature, adverse (positive energy) changes are more unlikely and hence, after most of the initial negative energy changes have relatively quickly been made, the process slows down greatly. At temperature zero no positive energy changes are allowed; labels can only be changed if they represent an immediate improvement in the segmentation. Figure 7-3 shows a sequence with T=0 starting with the randomly labelled initialisation. After about 2200 regions considered for relabelling, a steady state is reached in which three large regions remain. In this particular state no further negative energy changes are possible and therefore at T=0 this situation is absolutely fixed. Note that this does not contradict the results shown in Table 7-1; 2200 changes being less than the 35000 for T=3, because the real final state - all the same label - has not been reached and would only be so at zero temperature after an infinite number of changes considered.

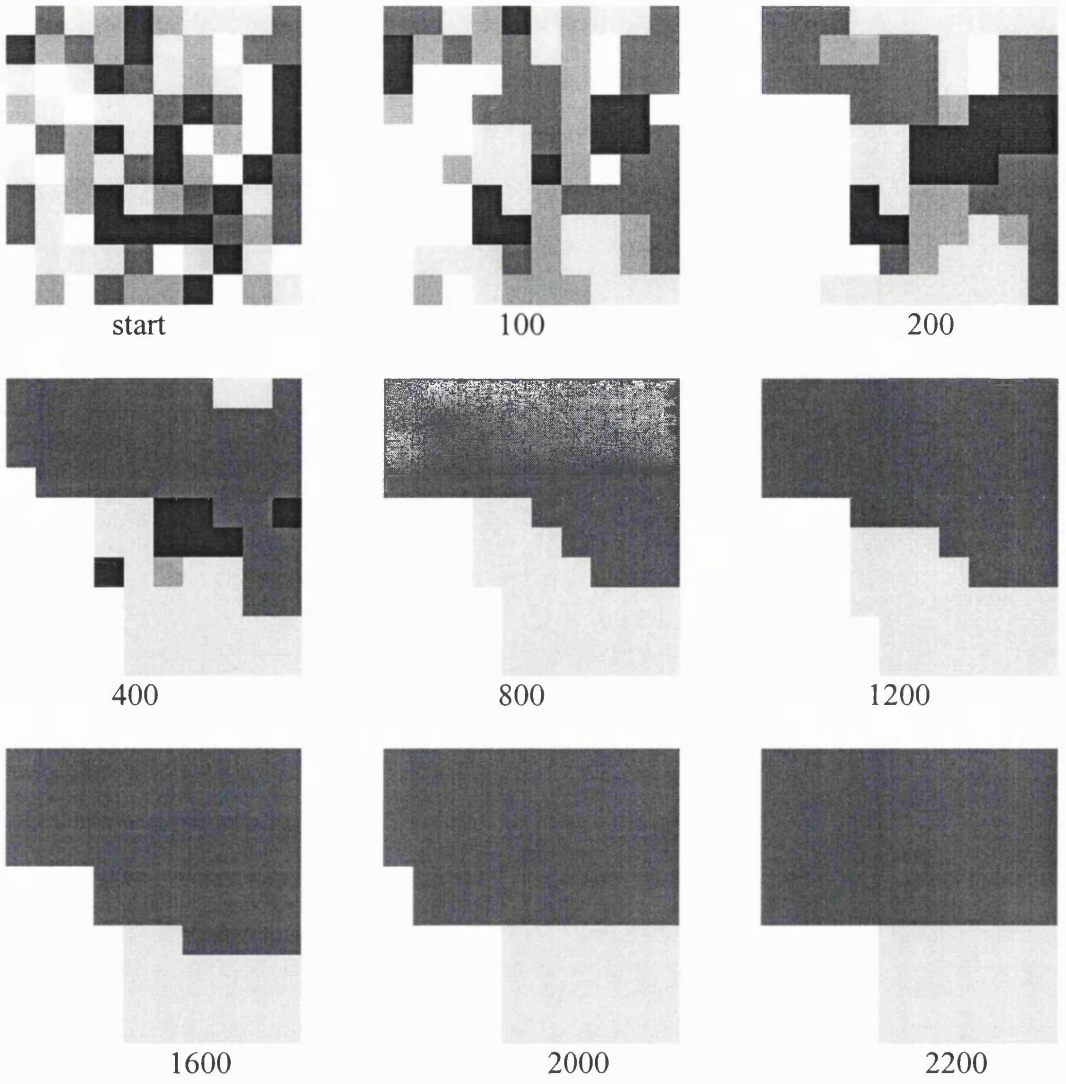


Figure 7-3 Annealing at T=0 with number of regions considered for relabelling

At all temperatures other than zero no segmentation labelling is fixed; any labelling change is always possible. However the likelihood of a positive energy change is increased as the temperature is raised. This is illustrated by the sequence shown in Figure 7-4 taken from the results at T=5. Here it is seen that although, after about 400-500 region-label-changes-considered, a similar looking segmentation stage is reached as for T=0, now the process continues to a proper final stage, eventually achieving an ideal segmentation after about 14000 changes considered.

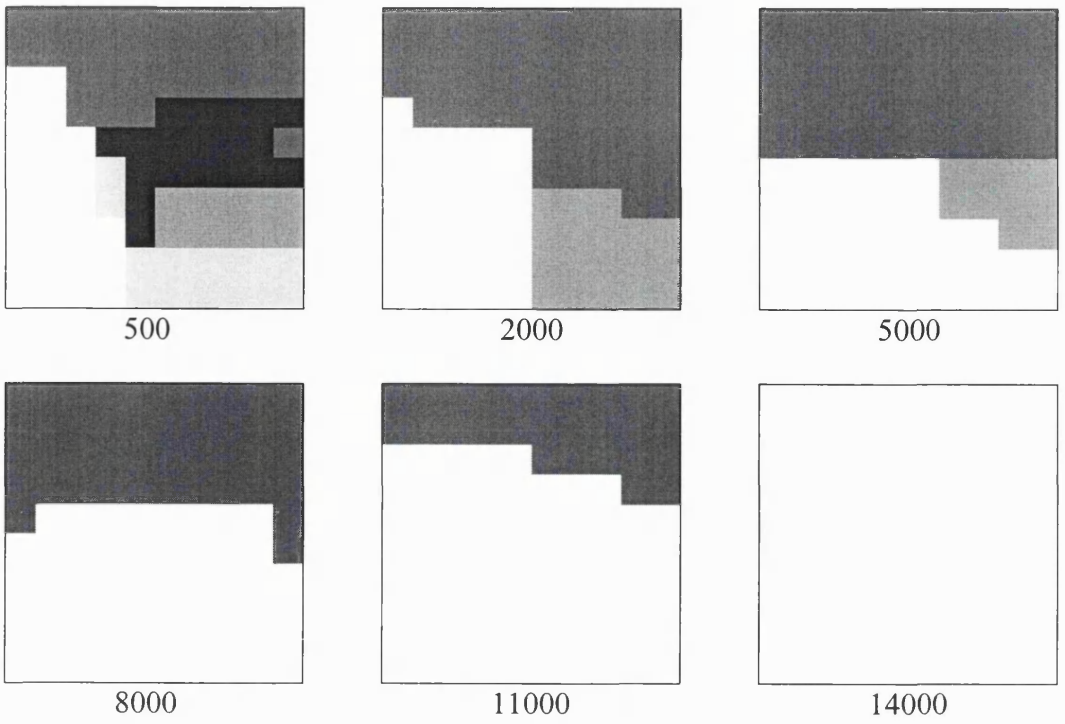


Figure 7-4 Annealing at T=5 with number of regions considered for relabelling

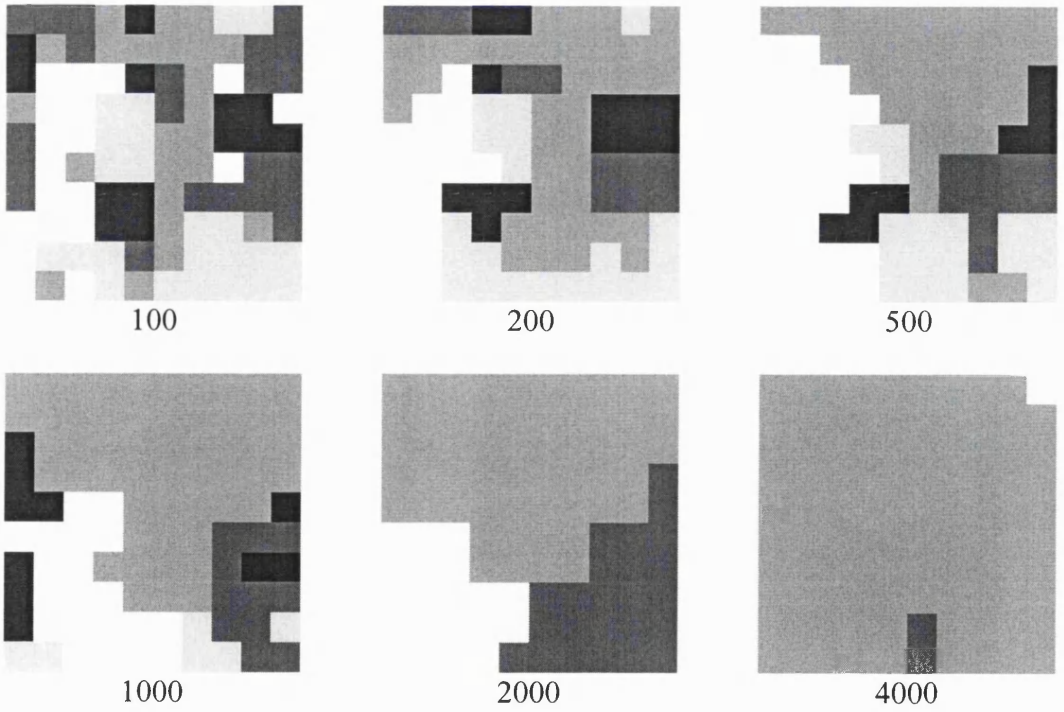


Figure 7-5 Annealing at T=7 with number of regions considered for relabelling

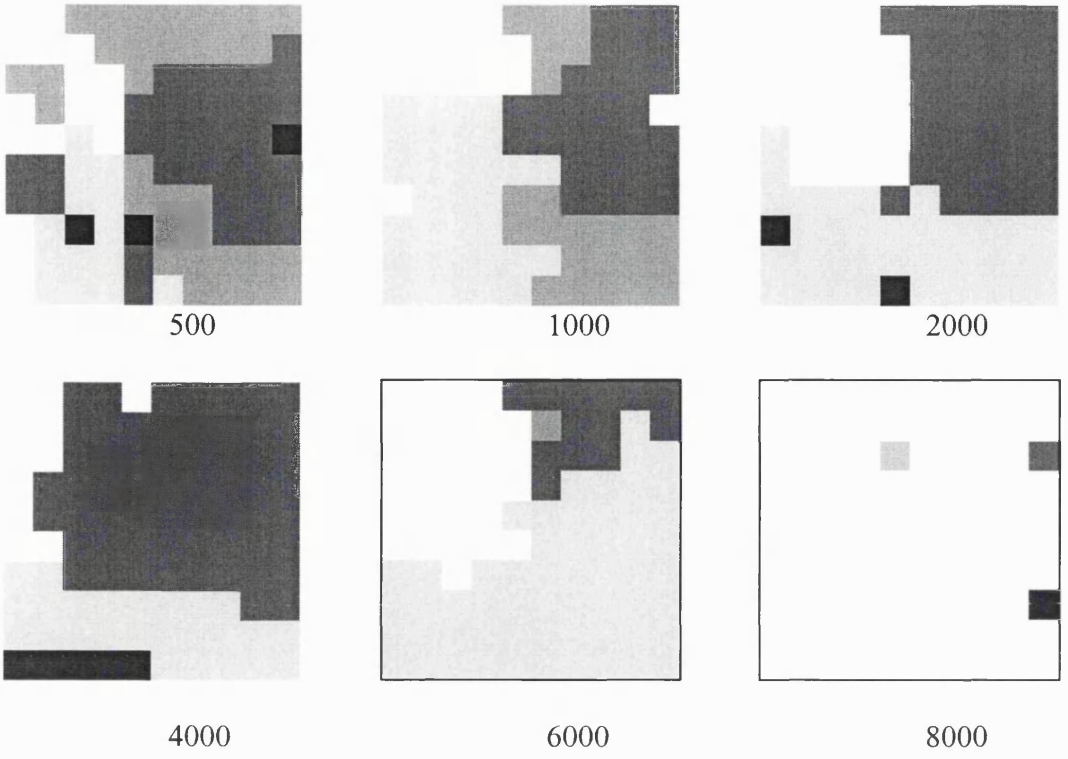


Figure 7-6 Annealing at T=8 with numbers of region changes considered

Two further sequences are shown in Figure 7-5 and Figure 7-6 for temperatures T=7 and T=8 respectively. The final states reached by both of these sequences contain mislabelled regions as already indicated in Table 7-1 and it is noted that these regions are not fixed but vary as the annealing is continued until the temperature is zero. Table 7-1 shows that the sequence for $T \approx 7$ is a sort of optimum as it requires a minimum number of region-label-changes-considered to reach the final state. Increasing the temperature further enables positive energy changes to be made more easily and hence blocks of labels can more readily alter but also this tends to happen more unnecessarily and the process takes longer to complete. However note that the image used has a highly artificial set of region boundaries containing no variation in boundary strengths and has an extremely simple ideal endstate so that it is unjustified to conclude that this optimum temperature will be the same for other more realistic images. Indeed a real image with different region sizes (and hence a range of boundary strengths) will necessitate a gradually reducing temperature (which is of course, as previously described, the whole basis of the simulated annealing technique); a temperature high enough to allow the larger regions to experience some positive energy change

relabellings will mean that smaller regions will be label changing far too frequently to enable a final steady state.

Even with the simple constant image, it can be noted from Table 7-1, that to achieve a perfect final state with the same label used over the whole image it is practically necessary to reduce the temperature to about 3-5 and theoretically to T=0. In an attempt to find the robustness of the annealing process applied this way experiments were made using different annealing schedules and those shown in Table 7-2 all achieved perfect results.

Starting Temperature	Step in Temperature	Number of steps	Number of region changes considered at each step
7	0.5	8	3000
7	0.5	8	4000
7	0.5	8	5000
7.5	0.5	9	3000
7.5	0.5	9	4000
7.5	0.5	9	5000
8	0.5	10	3000
8	0.5	10	4000
8	0.5	10	6000

Table 7-2 Successful annealing schedules for constant grey level image

Table 7-2 indicates that a consistent result is achieved providing that a sufficient number of region changes are considered within a certain range. The first row in the table might be regarded as a minimum kernel which needs to be applied and all those following are simply extensions of this kernel made by increasing the temperature range and/or by increasing the number of regions considered at each temperature step.

Now consider this annealing process applied to an image having true regions as shown in Figure 7-7. The image, like the previous constant image, contains 100^2 pixels and consists of four regions each constant in grey level so that the beliefs of connection/disconnection are strong indicators of the true boundaries. The regions are so disposed as to demand the use of four labels (the theoretical maximum) as each region is in contact with the other three.

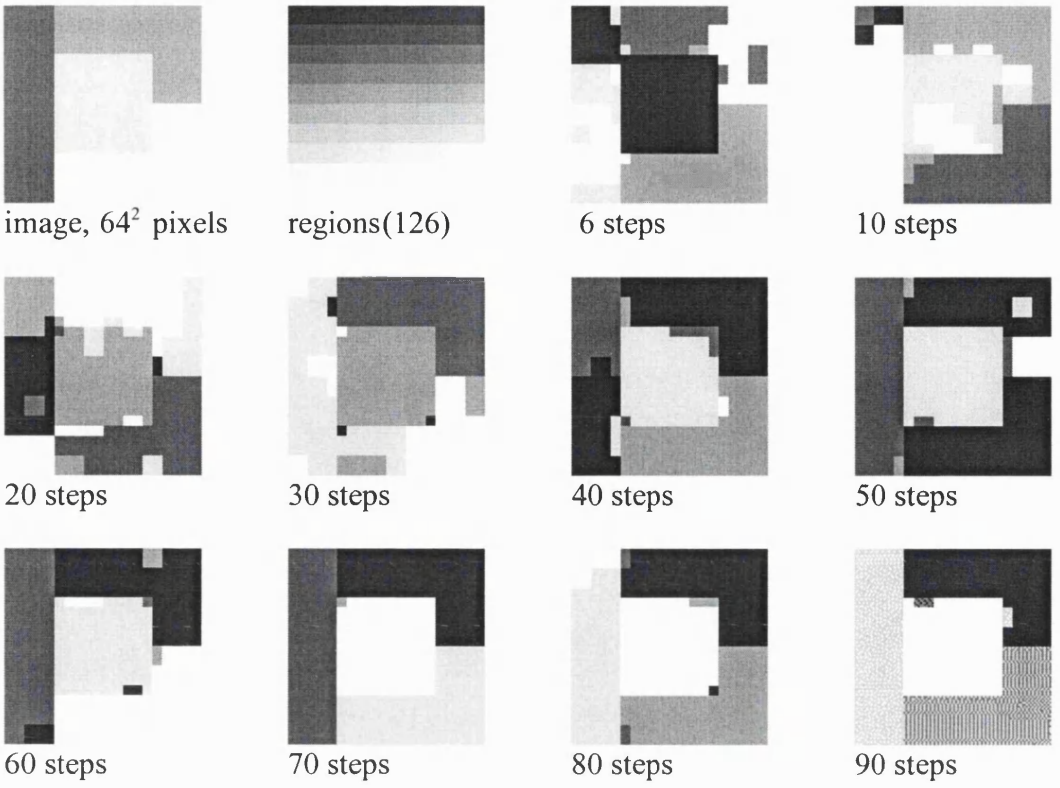


Figure 7-7 Annealing Stages: $T_0 = 9$; 90 steps of $\Delta T = 0.05$ ($T = 9 \rightarrow 4.45$); 10,000 changes considered at each step

The beliefs (thresholded at zero) are then segmented within sub-images of 10 by 10 pixels. It should be noted that now, unlike in the constant grey level image, the region sizes vary; the largest are still the sub-image windows (10 by 10 pixels), but, because these sub-images do not align exactly with the true region boundaries, there are now smaller regions giving a total of 126. The annealing sequence starts at a temperature of 9 and reduces in 90 steps of 0.05 to 4.45. At each step 10,000 region label changes are considered. It is evident that, even after only 5 steps (to T=8.55), large areas of the true regions are already blocked out. However the amount of blocking does not appear to alter significantly until about T=6.5; after about 50 steps. Furthermore during this stage the labels of the blocks change quite readily; note the grey level of the centre region in this period. At this temperature the regions are almost fully formed and their labels tend to stay constant although changes are still possible; note the change in labels of two of the regions from the 70 to 80 stage steps.

In Figure 7-8 another sequence is shown using exactly the same parameters. The only difference from that shown in Figure 7-7 is in the seed for the random number generator used to choose the next region for relabelling consideration.

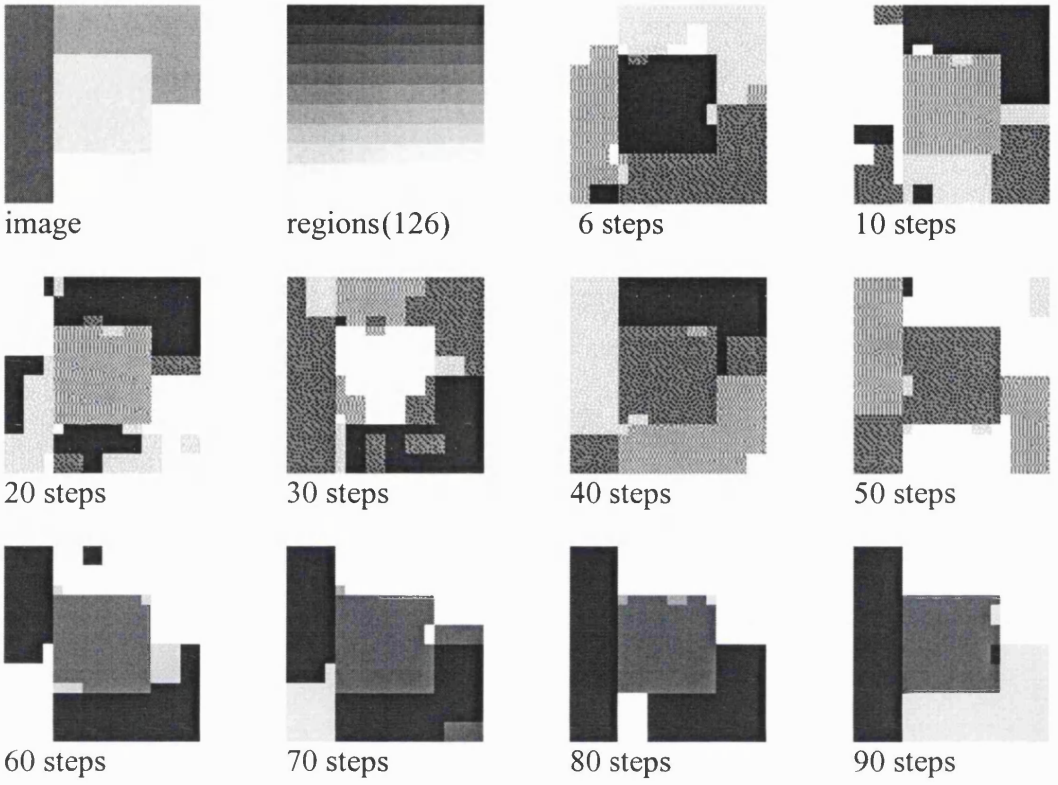


Figure 7-8 Annealing Stages: $T_0 = 9$; 90 steps of $\Delta T = 0.05$ ($T = 9 \rightarrow 4.45$); 10,000 changes considered at each step (different seed to previous result)

As a consequence of this change of seed the detailed changes of the whole sequence are completely altered but the end result is the same with the four regions correctly labelled differently. Here however it can be seen that, unlike the sequence shown in Figure 7-7, the correct end-point is appreciably slower in arriving. After about 60 steps the main labels used are incompatible; that for the left hand region is split between those used for the upper and lower regions and hence neither can be satisfactorily used for the whole region. Even after 80 steps (T~5) the whole of the lower region is relabelled in the following 10 steps.

In both of the sequences shown the process is not fully completed as a few regions remain incorrectly labelled. However as the temperature is reduced further the main state of the labelling now stays constant and these remaining regions are eventually

relabelled correctly and absolute constancy of labelling guaranteed when zero temperature is reached.

This relabelling of blocks of regions is felt to be very similar to the actual annealing of real materials: as the temperature is reduced large crystals are formed but the directional alignment of neighbouring crystals will be different; this is equivalent to neighbouring blocks of regions being labelled differently when ideally they would have the same label if the beliefs indicate they should be connected.

At the boundaries of these crystals atoms do not align with their neighbours on the other side of the boundary and hence are not in the lowest possible energy level. If the temperature is still sufficiently high then it is possible for atoms to change their alignment. This is more likely to happen at crystal boundaries, as here the energy change required is lower. In fact it may be only the first atom that requires a positive energy change and then its neighbours also at the boundary and their neighbours and so on and so on can all realign with neutral or negative energy changes, eventually moving the whole crystal boundary forward one atom.

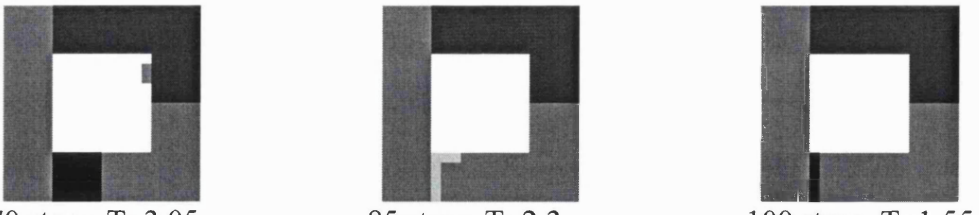
It is possible for an atom internal to a crystal to realign but this is bound to be a high positive energy change and therefore improbable and further it is much less likely to be useful in that the new direction of alignment will not necessarily correspond to any neighbouring crystals alignment. In this way, providing the annealing process is done sufficiently slowly within the optimum temperature range all the crystals will eventually join to form one single one.

This natural phenomenon appears to be very similar to the observed process of annealing the segmentations: at a low enough temperature relabelling of regions within blocks of like-labelled regions becomes very improbable. The *false* boundaries between differently labelled blocks of regions within a single real surface tend to move by the relabelling of regions at the boundary and the boundaries gradually move, row by row of regions until a true surface discontinuity is encountered.

This analogy breaks down in the presence of true region boundaries as then the minimum energy state is one with different labels present whereas a perfect, minimum energy, crystal will always have *all* its atoms aligned the same way with no dislocation boundaries.

The two sequences shown in Figure 7-7 and Figure 7-8 were among many tried using different seeds for the random number generator and all achieved the correct final result. As already noted, the starting temperature of 9 appears to be unnecessarily high as no significant changes seem to become apparent until the temperature is reduced to about 6. Further program runs were tried using a starting temperature of 6.5 with 40 steps of 0.05 reducing to the same temperature as before of 4.45. Again, at each step 10,000 region label changes were considered. These parameters also produced ideal results. The temperature cooling schedule is discussed further at the end of this chapter.

Perfection is not guaranteed, however, and, if the temperature gradient is not sufficiently low in terms of the drop in temperature number for a given number of region changes considered, then imperfections do arise. Figure 7-9 shows one result which occurred when only 2,000 region changes were considered instead of 10,000 at each step. As in the sequence shown in Figure 7-8, a stable solution was not reached when the temperature approached T=5.5 (there after 70 steps; here, starting at 6.5, after only 20) and the labelling underwent more large changes as the temperature was further reduced. Here the final segmentation, which did not alter further, was only reached after 100 steps to T=1.55 and, as can be seen, is not the ideal solution as the lower region is split into two segments one of whose labels could not satisfactorily be used for the whole region.



70 steps, T=3.05 85 steps, T=2.3 100 steps, T=1.55

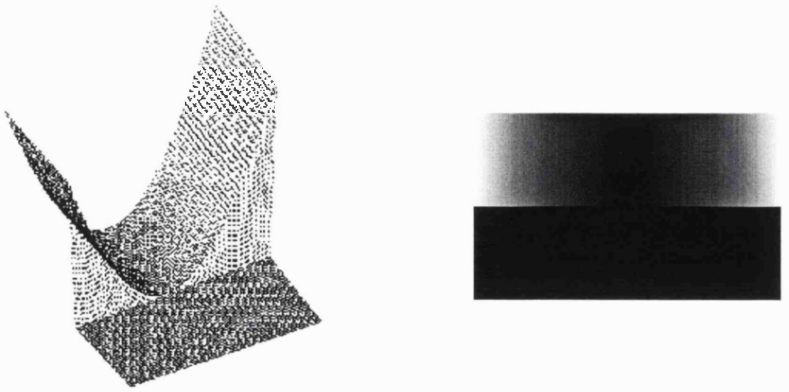
Figure 7-9 Annealing Stages: $T_0 = 6.5$; 100 steps at $\Delta T = 0.05$ ($T = 6.5 \rightarrow 1.55$); 2,000 changes considered at each step

This faulty result is not necessarily always a major problem as, if the labelled regions were now to be numbered there would be five instead of the ideal four, but a simple merging procedure using the original beliefs of connectivity would easily join the

lowest two regions. The first priority is to find all *real* boundaries; a few extra false boundaries may well be of no significance whatsoever as this example shows.

The two previous constant grey level and 'four-region' images were used to obtain information on the basic operation of the simulated annealing method and neither of them contained the actual defect of an incomplete boundary for which the sub-image technique was devised.

The image shown in Figure 7-10 was simulated to contain the problem in a somewhat idealised form. It consists of two areas; a quadratic surface abutted to a constant grey level surface. Both the grey levels and a three-dimensional representation are shown. The quadratic was converted to a one-byte integer format from an original real form ($y = 0.102x^2$) and as a consequence is no longer an exact quadratic but has the characteristics previously discussed (chapter 4). At the minimum of the quadratic it is in fact flat with exactly the same values as the constant surface. Hence, by design, the two surfaces are locally very different at one part and very similar (in fact exactly the same) at another.



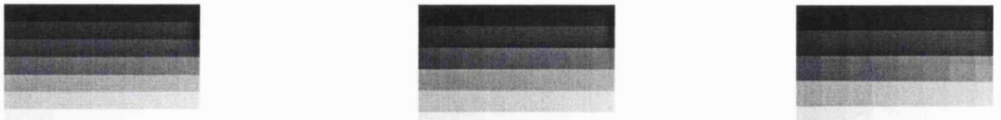
Quad-flat image

fitting beliefs $C_{DN} - D_{DN}$, ↑ $C_{DN} - D_{DN}$, →

Beliefs using $\sigma = 4$, $\alpha = \beta = 1.2$, $\gamma = 0.8$

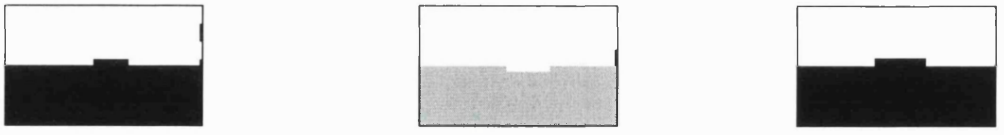
-0.963 -0.788 -0.262 0.140 0.558 0.879 1 1 1 1 1 1 0.558 0.140 -0.262 -0.579 -0.963

$C_{DN} - D_{DN}$, ↑ values along image discontinuity



9^2 (93 regions) 11^2(67 regions) 13^2(46 regions)

segmentations using shown size of sub-image



simulated annealing results; T_0 = 6.5, 60* ΔT=0.05,
5,000 regions considered at each temperature

Figure 7-10 Annealing results with quad-flat image

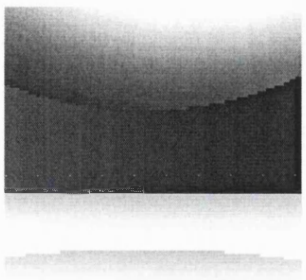
The fitting beliefs are shown in Figure 7-10 and indicate some fluctuation in the quadratic surface because of the imperfections but are everywhere greater than 0.99. The differenced beliefs of connection/disconnection are everywhere +1 in the flat surface and very high, >0.95, in the quadratic but at the boundary they change from the general value of ~ -1 to +1 at the centre. Only the values for the upwards direction are shown, these being the more interesting for this image but, of course, both directions are used in obtaining the actual results shown. The actual differenced belief values are shown for the boundary line at the centre; further out they approach -0.99.

Sub-image sizes for segmentation of 9^2, 11^2 and 13^2 pixels were chosen and the same annealing schedule used for all of them although, in principle according to the analysis given later, because of the different numbers of regions and their different sizes, both the starting temperature and the temperature gradient could have been more precisely engineered. Ideal results were obtained for this simple image and this was the case for many runs using different seeds for the random number generator used in region choice.

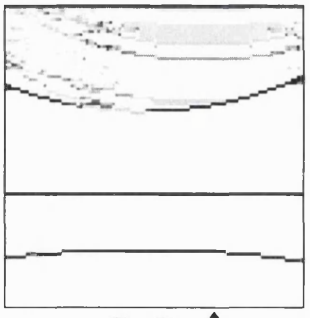
As a result of the true quadratic-flat boundary not exactly co-inciding with the sub-image grid (a deliberate consequence of the chosen sub-image sizes) the eventual best boundaries include a portion of sub-image boundaries as previously described (see Figure 7-1). The smaller the sub-image size chosen the closer the annealed boundary can approach to the true one. The adverse corollary of this is that a smaller sub-image size necessarily means a larger number of regions and a consequent increase in the number of labelling changes that need to be considered. In the limit individual pixels could be chosen as the sub-images and this would allow of a perfect result but to do this over the whole image might well be impractical (but see section 7.2)

The results of this technique used on the 'ball' image are shown in Figure 7-11. The surface fitting parameters are the same as those used in the previous chapter; Fig 6-2. A threshold of 0.2 is now used which results in a fairly clean boundary extraction but the ball boundary is incomplete. The segmentation is then performed within sub-images of size 10^2 pixels. This results in 135 regions, many of which are clearly the sub-images themselves. This segmentation is then annealed with a starting temperature of 6.5. All the major regions are found with the ball successfully completed using parts

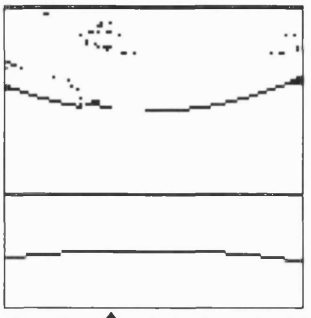
of some of the sub-image boundaries. This result was consistent, being repeatable with many different random number generator seeds.



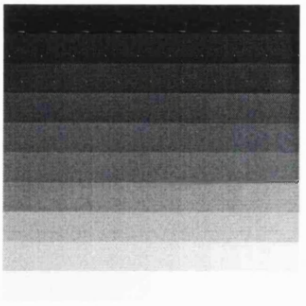
'ball' image, 100^2 pixels



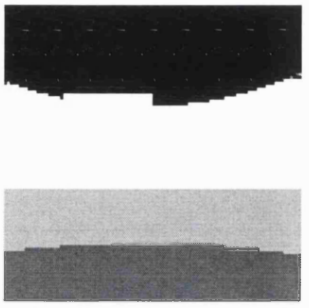
C_{DN}-D_{DN} ↑,



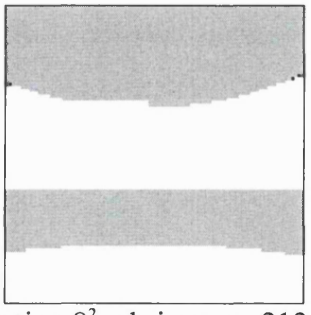
C_{DN}-D_{DN} ↑; 0.2 threshold



Segmentation
10^2 sub-images; 135 regions



Annealing: T_0 = 6.5; 82 steps ΔT =0.05;
6,000 changes each step



Segmentation 8^2 sub-images, 213 regions,
T_0 = 5.2; 80 steps ΔT = 0.05; 8,000 changes each step

Figure 7-11 Annealing results on ball image

Another result is shown in Figure 7-11 using 8^2 pixel sub-images. Here a starting temperature of 5.2 (= 6.5 * 8/10 - see the section on the cooling schedule at the end of the chapter) was used and the number of region labelling changes considered at each temperature increased to 8000 from the 6000 used in the previous runs. Again the ball boundary is successfully completed; this time with a visually closer approximation to what might be taken to be the *true* boundary.

7.2 Simulated Annealing applied to single pixels

The previous section showed how the problem of incomplete boundaries could be resolved by performing the segmentation within sub-images. This enabled boundaries with gaps to be completed using parts of the sub-image edges. However any boundaries so completed would be very unlikely to correspond to the true optimum boundaries as specified in the definition of the best segmentation. This would only be the case if these parts of the best boundary happened to coincide exactly with the sub-image grid - a clearly unlikely occurrence.

A closer approximation to the best boundary could be made by choosing sub-images of smaller dimensions as was demonstrated in the example given in Figure 7-11. Taking this idea to the limit the sub-images could be chosen to be single pixels and then the annealing (if done ideally) will indeed produce the *best* segmentation. The problem with this is that the number of regions for possible relabelling at each stage is then equal to the number of pixels and, although perfectly sensible theoretically, this is not a practical possibility.

A compromise is possible by performing the single pixel annealing only at those places where the results of the previous method (annealing of regions from sub-image segmentations) is not optimal; where the boundaries found correspond to the sub-image grid.

This is demonstrated in Figure 7-12. The first image is the annealing result made using 10^2 sub-images on the 'ball' image as previously shown in Figure 7-11. Using this it is simple to obtain those pixel-neighbour pairs where there is a change in labelling. This is shown only for the upward direction. Comparison of this with the original sub-image grid shows where the grid has been used as part of the segmentation. In practice it would then only be necessary to choose those sub-images which have part of a boundary affected for re-annealing on a pixel basis. Here is selected a somewhat larger than necessary region, 60 by 100 pixels, indicated in outline in the first image, to better illustrate the technique.

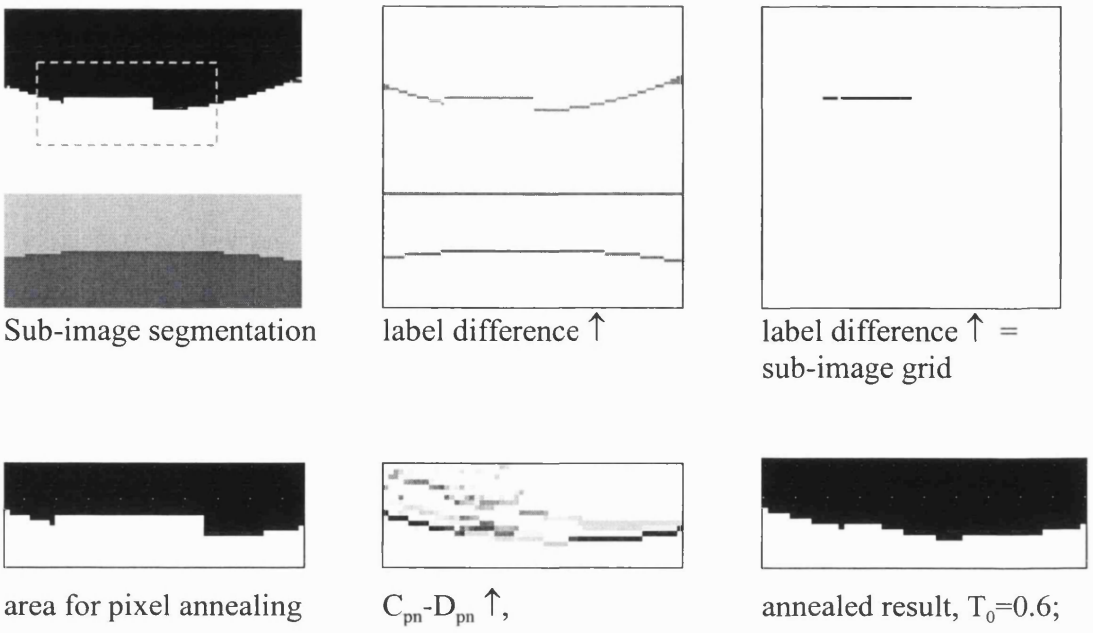


Figure 7-12 Single pixel annealing of a chosen area

Single pixel label annealing is now performed on this chosen region. The starting set of labels is taken as those currently in place from the previous method as reason would suggest that this is far preferable to a random initialisation. The original belief values are used but now there is no need to calculate the accumulated region boundary strengths: each boundary is a single pixel-neighbour junction and each C_{pn}-D_{pn} value itself a boundary strength.

Although this optimisation is now done in isolation from the rest of the image it is essential that the end-result be re-insertable into the previous labelling. For example, the two regions in the selected area happen to have labels 1 (top) and 3. It would be inappropriate to allow these to change to 0 and 4 (say) as, on reinsertion, the area would create two new artificial regions. Avoidance of this problem is made by the simple expedient of fixing the colour labels of the pixels on the outer boundary of the selected area thereby forcing the eventual labelling to conform to its surroundings as indeed it should. This has the added advantage of restricting the ideal segmentation result within the selected area to a smaller possible set: for the example shown only one labelling is possible whereas with five labels and no such restriction 5*4=20 labellings would be possible. This restriction should enable the annealing to reach an end result with a reduced number of labelling changes.

For the example shown, which consisted of 60*20 = 1200 pixels a starting temperature of 0.6 (slightly less than 6.5*1/10 - see later section - as, in principle, the initial labels should already be fairly near to the ideal) is used and reduced to 0.05 in 23 steps of 0.025. At each step an average of 200 labelling changes per pixel were made (200*1200, = 240,000 in total, at each step). This schedule gave consistent results over a number of runs.

Note that the annealed result closely follows the strongest (= most negative) parts of the belief values as is to be expected.

7.3 Annealing Temperature Schedule

The results obtained so far and in particular those with the simple constant grey level and 'four-region' images enable some predictions to be made on the appropriate choice of parameters of the annealing schedule.

There are three parameters involved; the starting and final temperatures and the gradient of numbers of region labelling changes versus change in temperature. Each of these is discussed below:

1. The starting temperature

It has been seen that there is no point in starting at too high a temperature; the region labels fluctuate rapidly to no advantage. It is the largest size regions which govern the optimum starting temperature. In the sub-image method of segmentation these are almost certainly going to be whole sub-image regions and many of these will have high connection values to their neighbouring sub-image block. It has been seen how very early in the annealing process large parts of a true surface will be covered by groups of like-labelled sub-image blocks and that the label of these parts taken as a whole needs to be able to change by a gradual relabelling starting at one edge. There seems to be little purpose in allowing changes internal to these areas as, firstly, the label so chosen would be random and therefore will not necessarily be compatible with the surrounding areas and secondly, as already stated, the amount of label changing is simply too great and not helpful to the process. These sub-image regions therefore need to have a sufficient probability of changing *at the boundaries of blocks of like labelled regions* to that of a neighbouring block.

As a way of gaining an insight to this an analysis is made of a particularly simple example which is not intended to be generally representative but merely to provide a helpful illustration. Consider the following figure which represents an early stage in the annealing process. The top horizontal solid line represents either the image edge or a true region boundary with an already correctly (= differently) labelled block. Assume

	0	0	**X**	1	1	1	1	
	0	0	**Y**	1	1	1	1	
	0	0	0	1	1	1	1	
	0	0	0	1	1	1	1	
	0	0	0	1	1	1	1	

that the vertical dashed line between blocks of like-labelled sub-image regions is a false boundary which ideally will be removed; the beliefs of connectivity are given by C_{pn}-D_{pn}=1 and the regions on either side should be all the same label.

Assume that the regions 'X' and 'Y' are currently labelled '0'. If the sub-image region size is w^2 pixels then the current energy of region Y is +w (three of the neighbouring regions are like-labelled, one differently) but on changing the label to 1 this becomes +3w a positive change of energy of +2w which will therefore be accepted with a probability exp(-2w/T). Similar reasoning shows that the probability of changing the label of region 'X' is given by exp(-w/T). These probabilities are governed by the ratio w/T and from the experiments described is known that good results are obtained with a starting temperature of T=6.5 for w=10 pixels. This gives probabilities of 0.046 and 0.215 for the 'Y' and 'X' types respectively. It would seem reasonable that the same probabilities would work whatever the size of sub-image regions chosen and that therefore a near optimum starting temperature is given by; T = 6.5w/10 and this result has been used in the previous work.

Implicit in this analysis is the assumption that within true surfaces the boundaries of neighbouring sub-regions have the maximum possible (negative) strength of -w arising from all the pixel-neighbour pairs having strengths D_{pn}-C_{pn} = -1. This may not be the

case; the beliefs of fitting and of surface similarity may well be less than +1 for reasons discussed in chapter 4. In such cases the starting temperature could be lowered but there is no necessity to do this: an unnecessarily high starting temperature is wasteful but not harmful.

2. Final temperature

Imagine a situation towards the end of the annealing process in which a region can change to a different label with only a tiny positive change in energy. The region's label will therefore be changing from the ideal to another very readily even at low temperatures. As there is no minimum energy change possible (it is *not* simply equal to the minimum boundary strength) the temperature needs to be reduced to zero for absolute confidence that no further labelling changes will take place. However, by monitoring the actual number of changes that take place at each temperature it might be possible to stop before this, either when no changes took place at the last temperature or when the only changes that did take place are deemed insignificant in that, for example, the size of region relabelled was unimportant.

3. The gradient of number of region labelling changes at each change in temperature

First an estimate of the total number of region *re-labelling changes* that need to be *considered* (rccs) in order to reach the optimum solution is made. Clearly this will be strongly dependent on the number of regions present. The arrangement of these regions will also have an effect: the number of neighbours that regions have on average. The distribution of boundary strengths might also play a part; as already noted the strongest and weakest strengths determine the range of temperatures that need to be used and it is possible that the spread of strengths will affect the average number of labelling changes per region that is needed.

As the simplest case consider the constant grey level image split up into equal size sub-image regions used previously. Here the boundary strengths are all the same and as shown before the temperature range needed to successfully produce the correct result (all sub-images labelled the same) is very narrow; in fact it is possible to anneal at a fixed temperature with the exception of a few random mislabelled regions remaining at the end. It has already been noted how the process consists of a succession of gradually increasing *crystal* sizes culminating in one of the size of the image itself.

Starting with an image of r regions define the number of region labelling changes considered needed to complete the process to be f(r). This means that an average of f(r)/r rccs per region will result in a crystal size of r. If the image is considered to be just part of a larger one then it seems reasonable that this average number of rccs per region will still accumulate crystals of this size.

Now consider halving the sub-image size making the total number of regions 4r. Applying the same average of f(r)/r rccs per region (a total of 4rf(r)/r = 4f(r)) should produce about four crystals each of size about r regions. Applying the same average number again should allow the crystals to completely relabel and thereby enable pairs of neighbours to adopt the same label. This would result in the stage with crystals of size 2r and, by the definition above, would be reached after f(2r)/2r rccs on average per region. Therefore:

$f(2r)/2r = f(r)/r + f(r)/r = 2\ f(r)/r, \quad \text{or} \quad f(2r) = 4f(r)$

The implication is that, $f(r) \propto r^2$; the total number of rccs needed to complete annealing is proportional to the square of the number of regions. In Table 7-1 it was shown that the number of rccs needed to complete a 100 region image at T=8 was on average 6600. If the sub-image region size is halved (to 5*5 pixels) then it would be anticipated that about 105,000 rccs (16*6600) would be needed. Ten trials with different random starting points required between 50,000 and 270,000 rccs with an average of 157,000 which is reasonable verification of what is only a rather ad-hoc theory.

Expanding this analysis to real images with the additional complications of different region sizes (and hence a range of boundary strengths) and with true boundaries one might as a first guess assume that the same number of rccs needs merely to be spread out between the necessary starting and final temperatures. This would mean, for example, that the four-region image demonstrated in Figure 7-7 with 126 regions would require of the order of 200,000 rccs. As stated previously perfect results were in fact obtained with 40 steps of 10,000 rccs - a total of 400,000 and it was apparent from the experimentation that a considerable amount of wastage was occurring with little happening between some temperature steps. It is conceivable that if the labelling changes were to be monitored in some way then the temperature reductions could be

automatically optimised and the number of rccs possibly reduced to the above estimate. Such a refinement was beyond the scope of this work

7.4 Summary

- It was first shown that a simple merging of the over-thresholded segmentation results produced in Chapter 4 would not necessarily be satisfactory and that a more sophisticated optimisation procedure is desirable.

- The use of four-colour labelling to record the results of the optimisation was demonstrated.

- A review of optimisation techniques was made and the problem put into a suitable format.

- The details of the method of simulated annealing as applied to the over-thresholded regions were explained.

- The reasons for a change to five (from four) colours were given.

- The method was demonstrated successfully on the 'ball' image and on a 'mushroom' image.

- The problem of boundaries with parts so weak they cannot be restored by over-thresholding was explained and the method of dealing with this by segmentation within sub-images followed by optimisation of the total set of regions produced described.

- Experimentation was made with constant grey level images to evaluate the process and gain some idea of suitable parameters to use. Further work was done with a 'four-region' image.

- Results were obtained using the 'quad-flat' and 'Ball' images and correct segmentations obtained.

- The method of refining the previous results by applying optimisation to the individual pixels of a subset of the segmentation was explained and results demonstrated.

- An analysis of the parameters used in the simulated annealing process was made including a first attempt to estimate 'correct' values.

8. Summary, Discussion and Recommendations

8.1 Summary

Segmentation is the division of digital images into regions. The final aim is scene segmentation in which the regions correspond to actual objects in the image; a car, a brain tumour, a flooded area. A more fundamental process is image segmentation, the subject of this thesis, which is entirely context free and results in regions which are homogeneous in themselves but may not necessarily correspond to whole objects. The basic assumption of this thesis is that all images can be segmented into regions that have certain consistent characteristics.

A region classification was defined which postulates two basic region features: firstly a 'smooth' grey level variation in which, as the name implies, the grey levels can have first order gradients of change of any magnitude but the changes in these gradients, the second order changes, must be relatively small. Secondly, overlaying the smooth variation, are textural variations which include 'noise'. This thesis is concerned mainly with the 'smooth' grey level variations and the problems they present for segmentation.

Some existing 'edge-detection' techniques rely directly on the detection of high second order gradient changes to define where region borders are present but then tend to ignore the degree of connectedness of other pixels with their neighbours. They also have a technical problem of deciding in which regions to place pixels classified as edges. As described in Chapter 3 some present methods use a surface fitting which forces whole regions to fit a particular arbitrarily chosen function often with no real justification. Few existing segmentation methods make any formal definition of the best achievable result but simply produce an end result when the algorithm finishes.

This thesis describes the calculation of the connectivity or otherwise of neighbouring pixels based on a local surface fitting method. A precise definition of the optimum segmentation based on these connectivities is made. Methods of producing segmentations approaching this ideal are specified, implemented and results demonstrated.

8.2 Discussion of Achievements

A novel method of calculating the degree of linking of neighbouring pixels dependent on the similarity of their associated best measured local surfaces was described. First a local surface fitting method using a biquadratic function in 5 by 5 windows was defined in which, for each pixel, the best window was chosen as that with the lowest r.m.s error of fitting. The mathematics of the fitting method were explained showing that the coefficients of the fitting are given by the product of a 'pseudo-matrix', constructed from the coordinates of pixels within the window, with a vector of the pixel grey levels.

A 'belief' of fitting was defined for each pixel as an exponential function of the r.m.s error of the best surface giving values between 0 (poor fitting) and 1 (perfect fitting). A belief of surface similarity for the surfaces of each pixel-neighbour pair was also defined. Again an exponential function was used with the exponent this time being a weighted linear sum of the differences between the coefficients of fitting; the coefficients having first been expressed in terms of the same coordinate axes. The beliefs of pixel-neighbour connection and disconnection were then calculated in terms of the previously defined beliefs.

The strength of a segmentation was defined as a sum, taken over all pixel-neighbour pairs, of either the belief of connection or of disconnection. Where the pixel-neighbour labels are the same the belief of connection is included, otherwise the belief of disconnection is used. The optimum or 'best' segmentation was then defined as that with the greatest strength and it was shown how this definition parallels the intuitive idea of a 'good' segmentation by completing boundaries even where they are locally weak.

A method of thresholding the beliefs of connection-disconnection was developed to produce a set of links from which a segmentation can be formed. The zero-thresholded segmentation was shown to be a subset of the best segmentation. Over-thresholding and the consequent effect of producing more non-linking and an increased number of regions was described.

The thresholding technique was applied to a variety of surface types including an ideal test image consisting of exact biquadratic surfaces plus a simulated 'ball' image and real mushroom images to demonstrate the qualities and problems associated with this technique. In particular the difficulty of surfaces with partially covered windows producing low fitting beliefs and hence undesirable non-linking with consequent multiple small regions, was illustrated and the effects of quantisation of images to a one-byte resolution was analysed and demonstrated.

The segmentation technique developed in this thesis was compared with one based on Markov random field (MRF) optimisation. This comparison demonstrated the weakness of the MRF approach which, based on knowledge of pixel intensities alone, was unable to distinguish true boundaries as defined by changes in gradient.

Next the problem of producing superfluous regions from 'over-thresholding' was addressed. It was first shown that a simple merging of the over-thresholded segmentation results is not necessarily satisfactory and that a more sophisticated optimisation procedure is desirable. A review of optimisation techniques was carried out and the problem expressed in a suitable mathematical format. The use of four-colour labelling was demonstrated as an efficient means of recording the results of the optimisation. The method of simulated annealing was described and the details given of its application to a set of over-thresholded regions. The technique was demonstrated successfully on the over-thresholded results of both synthetic and real images.

Over-thresholding cannot solve the problem of region boundaries having some parts which are not merely weakly disconnected but actually connected. Chapter 7 showed how this problem can be partly resolved by dividing the image into smaller sub-images, applying the initial segmentation procedure independently within each sub-image and then optimising the final set of regions. Complete boundaries can then be obtained from the sub-image edges. This use of image sub-division increases the number of regions and many subsequently need to be relabelled to be consistent with their neighbours. Considerable experimentation was made with both constant grey level images and a simulated 'four-region' image to evaluate the process and gain some idea of suitable parameters to use. Successful segmentations were then obtained using further, more complex images.

Although division into sub-images can successfully complete region boundaries by including parts of the sub-image boundaries, these are not necessarily part of the defined 'best' segmentation. The smaller the size of sub-image chosen the more nearly the best segmentation can be approached and, in principle the perfect result can be achieved if individual pixels are used as the starting point. However, in practice, this would result in an intolerable computational burden. A method was therefore developed for detecting the imperfect parts of a segmentation and which applied optimisation only to those regions at the pixel level. This led to improved results within an acceptable time scale.

8.3 Recommendations for Future Work

A difficulty with the method as implemented is that all pixels are treated in the same way irrespective of the quality of surface fitting. This limitation is especially noticeable when applied to a genuinely textured or very noisy surface region. In this case, the beliefs of fitting are very low and this results in the beliefs of connection and disconnection also having low (~0) values. Thresholding such beliefs (either zero or over-thresholding) results in a high degree of non-linking and a large number of single-pixel regions. This is both uninformative and disruptive of the subsequent optimisation processes which have to deal with an unnecessarily large number of regions.

This problem could be resolved and a more constructive output achieved if those pixels where the belief of fitting is low were treated separately. These could be grouped together, irrespective of their beliefs of connection-disconnection, into regions of noise or texture which could then be examined, if required, by a later process. This could be either the same procedure with different parameters to allow for a higher noise level or a different process which perhaps analysed on the basis of a texture measure.

A technical improvement which could be made concerns the calculation and storage of the region-neighbour boundary strengths in the optimisation program. Currently this is done correctly but inefficiently using matrices of the size; (number of regions)^2. As most regions only have a few neighbours these matrices are usually very sparse and hence using them consumes more time than is necessary. In particular, the program storage requirements become oppressive when thousands of regions are involved (which can be the case if the previous problem concerning noise is present).

Formulating a more efficient storage of this information using, for example, linked lists, would considerably speed up the program and enable larger numbers of regions and hence bigger images, to be processed. As an example, a segmentation with a thousand regions currently needs a million memory storage locations for region-neighbour strength information which, assuming an average of four neighbours per region, could be reduced to four thousand. Hence the maximum size of image would increase by a factor of about 250. Furthermore all regions are currently checked for possible adjacency and removal of this unnecessary processing would lower the overall time significantly (although by less than a factor of 250 as a considerable amount of processing would still be required for the adjacent regions).

As currently implemented the programs operate with a single fitting window size and a single sub-image window size dictated by the user. This formulation is not general as a multiscale version is possible using a series of window sizes. The modification described in Chapter 7 for processing the image as an array of sub-images, then detecting and relabelling where sub-image boundaries were incorrectly used as part of the segmentation, could be automated. The process could then be expanded to include a series of relabellings at gradually reducing sub-image sizes, stopping when a consistency is reached. The choice of parameter values for the process currently decided by the user including the noise level, (σ) and the three surface similarity parameters, could form part of the automation by making them dependent on detected noise levels. However, it seems likely that the user would still need to dictate his or her requirements as these have an implication on the 'correct' parameter choice.

More radically the production of the beliefs of connectivities itself could be iterated at different scales using different window sizes. The area of pixels selected for fitting could be made less restrictive than a rectangular window thereby allowing single badly fitted pixels found at an earlier iteration to be included in a different group at a later stage.

It is anticipated that these proposed changes could form part of a viable follow-on research project resulting in further improvements to the segmentation technique developed in this thesis.

A. Program Environment, Specification and Computational Cost

A.1 Khoros Programming Environment

All of the software written for this thesis has been produced using the Khoros environment which is briefly described here.

Khoros is an integrated software development environment for information processing and visualisation which was initiated at the Department of Electrical and Computer Engineering at the University of New Mexico (UNM) in 1987. The more recent development of Khoros is due to Khoral Research Incorporated, a company set up by the UNM for this purpose.

Khoros incorporates many features which are not used in this work and they will not be described here. A more comprehensive description can be found in Konstantinides and Rasure[87]. The version used here is Khoros 2.1. Khoros was originally intended for a Unix type operating system and that is what has been used in this work (a Sun/Unix based system at UCL, a SGI/Unix system at Sira and a 486/Linux system elsewhere) but a Windows NT version is now available.

The main usage of Khoros is via *Cantata* which is a visual programming language. Cantata is based on a data flow structure of connected programs. A typical system constructed using Cantata is shown in Figure A-13. Within Cantata programs are selected from a hierarchy of menus (totalling over 260 programs) covering many aspects of image processing, numerical analysis and graphical/visual display. The program 'glyph' is selected from these menus and placed on the Cantata 'canvas'. Connections between programs are then made by clicking on the appropriate input/output buttons of the glyphs. Glyphs communicate by the use of temporary files which are set up automatically when connections are made. In Khoros the entire workspace, the data flow and intermediate results can be saved and restored for later use.

Within Cantata programs are run simply by clicking on their central 'run' button. This is allowed only if previous programs in the hierarchy have already been run..

Alternatively the whole workspace may be initiated in which case the scheduling of different programs is automatically controlled.

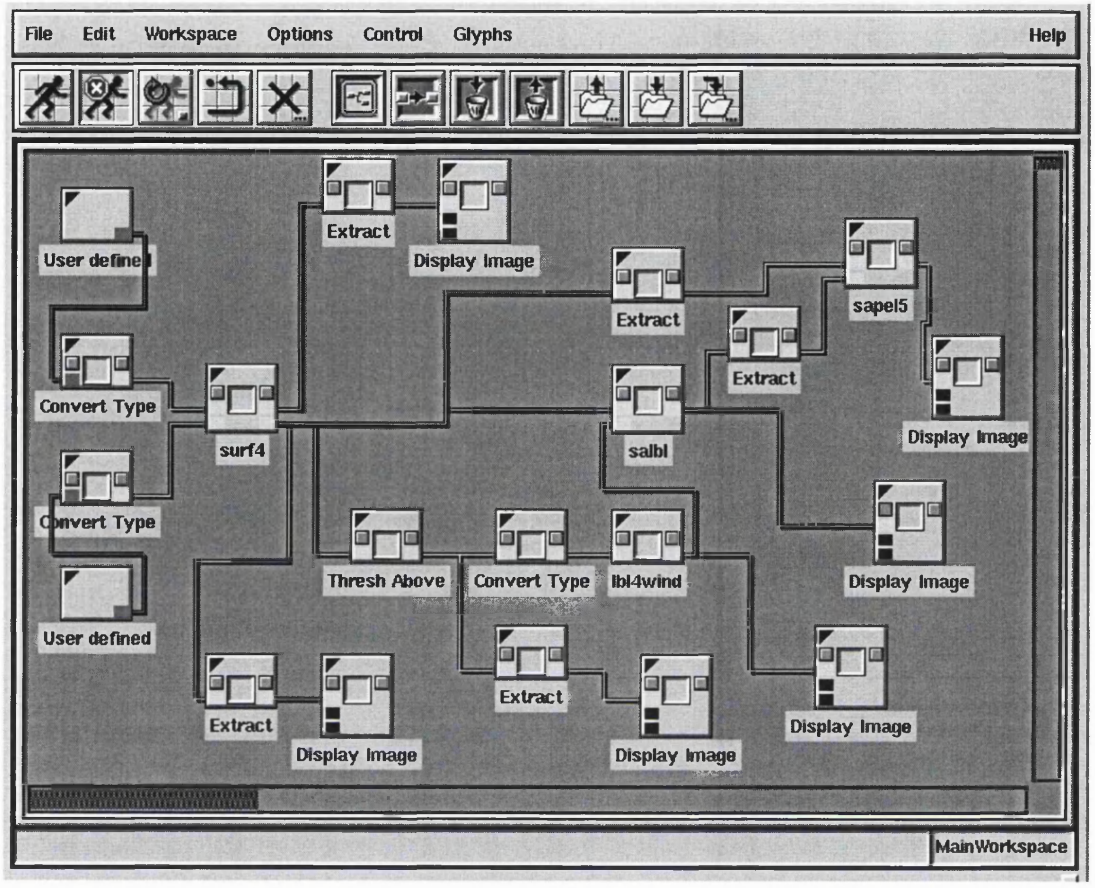


Figure A-13 Cantata Programming Environment

A particular advantage of Khoros for this work is the highly flexible nature of the data that can be handled; there is no restriction to two-dimensional images but rather a 'polymorphic data model' is implemented. The details of this model are complex and unimportant for our purposes but this meant that different items of data for each pixel could conveniently be stored and easily accessed with existing programs. For example the set of six coefficients calculated as the best fit polynomial was stored as a series of six planes in one data set.

Interaction with individual programs by the user can be made via the program 'pane' which is a window that can be opened by clicking on the correct glyph button. The pane contains information on the names of input and output files (as set either

automatically or by the user) and the values of input parameters to the program. An example of a pane for the program Salbl (see next section) is shown in Figure A-14 in which three i/o files are named and 5 parameters can be set.

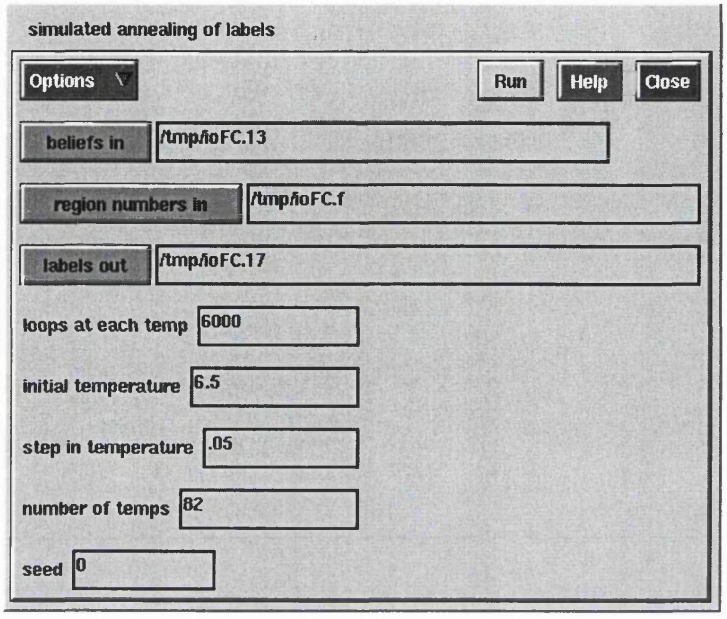


Figure A-14 Example 'pane' for the program 'Salbl'

Khoros also provides software to assist in the production of new programs for the Cantata environment. Designing the pane is the first step in writing a new program. Once the layout of the pane is planned; the input/output file information fully defined; user-input parameters data-typed and variable names chosen, an automatic code writer produces c-code for the data structures required. All (!) that remains is to complete the rest of the coding.

A.2 Program Specification

Figure A-13 shows a somewhat idealised workspace as an example of those used in this thesis. Inevitably, in practice, actual workspaces tended to become much more crowded with extra programs used temporarily. The programs in Figure A-13 which

were written by the author are 'Surf4', 'Lbl4wind', 'Salbl' and 'Sapel5' and all the other glyphs are as supplied with Khoros.

The data flow in Figure A-13 is mainly from left to right with vertical digressions (the inputs to all program glyphs are always on the left and the outputs on the right).

Surf4 performs the local surface fitting within five by five pixel windows as described in chapter 4. The left-most 'user-defined' boxes are not programs but simply definitions of the input data, one being the actual image to be segmented and the other the B-matrix used in the fitting process. This latter piece of information was constant and therefore could clearly have been incorporated into Surf4 but was, at least initially in the study, the subject of some experimentation and purely for convenience was kept separate. Both data inputs are first converted to type 'double precision' and then input to 'Surf4'.

Surf4 has two output data sets; the upper one contains the six coefficients of fitting and these can be individually examined by setting a parameter in the 'extract' program and then displayed. The lower output from Surf4 consists of three planes of data, again of the same size as the original image. These are the beliefs of fitting and the differenced beliefs of connection and disconnection in the vertical and horizontal directions. These beliefs can also be individually examined.

The differenced beliefs from Surf4 are thresholded, with a threshold chosen by the user and input via the 'threshold' program *pane*, to form binary link/non-link information and the result, after conversion to one-byte integer, passed to the program 'Lbl4wind'.

'Lbl4wind' is the pixel labelling segmentation program which takes the link information and accumulates groups of linked pixels and gives them the same region label to produce a segmentation. The Lbl4wind *pane* enables the user to set the window size within which successive segmentations are performed. By setting the window size less than the image size the sub-image segmentation technique described in chapter 7 can be applied. The output from Lbl4wind can then be displayed.

The segmentation region labels together with the original belief information is passed to the simulated annealing program, 'Salbl', which produces an optimised set of colour labels which can then be displayed. As the pane for Salbl, Figure A-14, shows, the user

needs to specify five parameters; the initial temperature, the step in temperature, the number of temperature steps, the number of region label changes considered at each temperature and the seed for the random number generator. The five colour label result can then be displayed

A subimage of the segmentation and of the beliefs can then be extracted and this information fed to the program Sapel5 which retains the labels of pixels on the outer boundary but optimises the labels of the inner pixels as described in Chapter 7.

These four main programs together with any functions written are described below in 'pseudo-code'. All detailed equations are already contained in the previous text and are not repeated here.

- **Program Surf4**

This program computes and outputs the differenced beliefs of connection/disconnection.

```
INPUT B-matrix file/ image file/σ, α, β, γ
COMPUTE Pseudo-matrix
SET rms error stored for all pixels to high value
FOR all 5*5 pixel windows
        COMPUTE coefficients of fitting;
   COMPUTE rms total error
        FOR all pixels in window
                IF (rms error < that already stored for pixel)
          STORE new rms error for pixel,
          STORE pixel coords relative to  window
       ENDIF
   ENDFOR
ENDFOR
FOR all pixels
   COMPUTE Belief of fitting
ENDFOR
FOR all pixels
        RECOVER pixel coords relative to best fit window,
```

```
        RECOVER Belief of fitting, coefficients
        FOR two neighbours - North and East
                COMPUTE neighbour coords relative to pixel
                IF (neighbour is part of image)
                        RECOVER neighbour coords relative to best fit window
                        IF (pixel window not same as neighbour window)
                                RECOVER neighbour Belief of fitting, coefficients
                                COMPUTE coords pixel and neighbour relative to midpt
of windows
                                CALL BELIEF_LINK(pel/nghbr coeffs, coords midpt,
σ, α, β, γ)
                        ELSE
                SET Belief of surface similarity =1.0
                        ENDIF
            COMPUTE Difference of beliefs of connection/disconnection
                ENDIF
    ENDFOR
ENDFOR
OUTPUT results
END
```

- **Function BELIEF_LINK**(pel/nghbr coeffs, coords midpt, σ, α, β, γ)

This function called by Surf4 calculates the belief of surface similarity

```
COMPUTE neighbour coefficients translated to midpt of two windows
COMPUTE pixel coefficients translated to midpt of two windows
COMPUTE difference of translated coefficients
COMPUTE belief of surface similarity
RETURN belief
```

- **Program Lbl4wind**

This program groups linked pixels together within the defined sub-images and gives them the next region label.

```
INPUT thresholded links, window size
SET current label = 0
FOR all window positions
   COMPUTE top-left (tl) AND bottom right coords (br) of window
   IF (br not in image)
            SET new br = br of image
   ENDIF
   FOR all pixels in window
            IF (not already labelled)
                  SET new current label = old current label +1
         SET label of pixel = current label

                  CALL NGBRS_LABEL(pel coords, window tl/br, image size,
current label,
                                    links, labels)
            ENDIF
   ENDFOR
ENDFOR
OUTPUT labels
END
```

Function NGBRS_LABEL(pixel coords, window TL/BR coords, image size,
current label, links, labels)

This function, called by program Lbl4wind, recursively labels the linked neighbours of the current pixel to the same label within the current window.

```
FOR (each of four neighbours)
      COMPUTE neighbour coords relative to image
      IF (neighbour within current window)
```

```
            IF (neighbour not already labelled)
                IF (neighbour linked to pixel)
                    SET neighbour label to current label
                    CALL NGBRS_LABEL(neighbour coords, ",",",",",")
                ENDIF
            ENDIF
    ENDIF
ENDFOR
RETURN
```

- **Program Salbl**

This program inputs the region labels and beliefs of linking and outputs a set of 'colour' labels optimised by simulated annealing.

```
INPUT beliefs of linking
INPUT region labels
INPUT Initial temperature, step in temp, number of temps, number labelling changes
        considered at each temp, seed for random number generator
INITIALISE matrix of neighbour indicators for each region at 0
INITIALISE matrix of region-neighbour boundary strengths at 0
FOR each region
    FOR each pixel within region
        FOR neighbours of pixel to North and East
            IF neighbour in image
                IF neighbour not in same region as pixel
                    SET region-neighbour indicator to 1 (yes)
                    INCREMENT region-neighbour strength by pel-
nbr belief
                ENDIF
            ENDIF
        ENDFOR
    ENDFOR
SYMMETRISE region-neighbour matrices to account for South, West neighbours
```

**********Simulated Annealing starts here**********

INITIALISE colour labels (0,1,2,3,4) of all regions randomly
FOR each temperature
    FOR 1 to Number of requested labelling changes to consider
        CHOOSE new random region
        INITIALISE energy of new labelling = 100000 (v high)
        FOR each of 5 possible colour labels starting at random colour
            INITIALISE region energy = 0 for this colour label
            FOR all neighbours of region
                IF neighbour has different colour label
                    INCREMENT region energy with region-neighbour strength
                ENDIF
            ENDFOR
            IF colour label different to region current colour label
                IF region energy < energy of new labelling
                    SET *new* energy = region energy
                    SET best new label = colour label
                ENDIF
            ELSE
        SET *current* energy = region energy
            ENDIF
        ENDFOR
        COMPUTE lowest possible energy change = new energy - current energy
        IF lowest possible energy change < 0
            SET region label = best new label
        ELSE
            COMPUTE probability of accepting energy change
            COMPUTE random number range 0 to 1
            IF random number < probability

```
                        SET region label = best new label
                    ENDIF
                ENDIF
    ENDFOR
ENDFOR
OUTPUT colour labels
STOP
```

- **Program Sapel5**

This program inputs the colour labels produced by Salbl and the original beliefs of linking (or subsets of the same). The labels of individual pixels within a one pixel border of the edge are optimised by simulated annealing.

The code is similar to, and somewhat simpler than, that of Salbl so it has not been reproduced here. The main difference is that, because the only neighbours, by definition, are the four nearest pixels, matrices for region-neighbour pairs and accumulated strengths are not needed.

A.3 Computational Cost

The following gives the computational times taken by the various programs as measured on a 486 machine at 66Mhz with 12MB of RAM using a Linux operating system.

- **Surf4**

The runtimes for the surface fitting program were in direct proportion to the image size which was to be expected as the same calculations were carried out at each window position. For a 128^2 pixel image the time taken was 108 seconds.

- **Salbl**

This program had two parts: an initial calculation of the boundary strengths followed by the simulated annealing procedure. It was the latter which dominated the runtimes and, for a given number of regions, these were in direct proportion to the product of the number of region relabellings at each temperature with the number of different

temperatures; that is, to the total number of relabellings. For a segmentation with 524 regions a total of 1,050,000 relabellings gave a runtime of 18 minutes and 17 seconds.

The runtime was also dependent on the number of regions, increasing roughly in proportion with them. This was almost certainly due to the considerable amount of searching for possible neighbours at each relabelling as already described in Section 8.3. This is evidenced by the fact that for the four region image used in Section 7 the runtime was 35 sec for the same number of relabellings as above (1,050,000).

- **Sapel**

Not needing the same storage or searching as Salbl resulted in this program being much quicker. A 23^2 pixel image with 400 average relabellings per pixel at each of 5 temperatures (a total of 1,058,000 relabellings) required 22 secs.

- **Lbl4wind**

This program was invariably much quicker than the others requiring less than 2 secs for completion of 128^2 pixel images.

B. Markov Random Fields

The use of Markov Random fields is based on Bayesian estimation which, applied to the task of image segmentation, is a form of classification in which pixels are assigned to given classes depending on the grey levels, possibly features calculated from them and perhaps including information known about the particular image type. Connected groups of pixels in the same class can then be given the same region label to form a segmentation.

Bayes's law states that the probability of any pixel having both a grey level, y and a class, x, can be expressed as two equal products:

$$P(y)P(x \mid y) = P(x)P(y \mid x) \qquad \text{Eq B-1}$$

The right hand side of Eq B-1 is a product of the $P(x)$, the a-priori probability of being in class x, with the $P(y|x)$, the class-conditional probability of having a given grey level, y, assuming the class, x, is given. The left hand product is of the $P(y)$, the probability distribution of the grey levels, with the $P(x|y)$, the a-posteriori probability of being in class x given that the grey level is y.

The task is to assign to all pixels their most probable labels x given a set of observations (*i.e.* data in the image) y; in other words to maximise the a-posteriori probabilities, $P(x|y)$ by correct choice of the pixel classes. From Eq B-1 it follows that,

$$P(x|y) = \frac{P(x)P(y|x)}{P(y)} \qquad \text{Eq B-2}$$

As the $P(y)$ are dependent only on the pixel grey levels they are fixed for a given image and hence maximising $P(x|y)$ reduces to maximising the product $P(x)P(y|x)$. Models for both the $P(y|x)$ and the $P(x)$ are therefore required.

Commonly the grey levels for all classes exhibit a spread and the class-conditional probabilities are often modelled as Gaussians:

$$P(y \mid x) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left[-\frac{(y-\mu_x)^2}{2\sigma_x^2} \right], \qquad \text{Eq B-3}$$

which demands only that the means, μ_x, and standard deviations, σ_x, for all classes, x, are known or somehow estimated (methods of achieving this are described later).

Using Eq B-2 and Eq B-3, maximising the $P(x|y)$ becomes equivalent to maximising:

$$\frac{1}{\sqrt{2\pi}\sigma_x} \exp\left[-\frac{(y-\mu_x)^2}{2\sigma_x^2} \right] P(x)$$

or;

$$\frac{1}{\sqrt{2\pi}} \exp - \left[\ln \sigma_x + \frac{(y-\mu_x)^2}{2\sigma_x^2} - \ln P(x) \right] \qquad \text{Eq B-4}$$

A simple model for the a-priori probabilities $P(x)$ would be to assume that they are equal to the proportion of the class which is present. For example, if it was known that the objects occupied 10% of the image in a two class system then P(object) = 0.1 and P(background) = 0.9. These models would enable the maximisation of the $P(x|y)$ for each grey level and hence each pixel, independently, thereby securing maximisation for the whole image.

A more sophisticated model for the $P(x)$ might be made if it is thought that the class for each pixel is affected by the classes of its neighbouring pixels. This might well be if, for example, the objects are known to have sizes of the order of a few or perhaps many pixels, when the probability of an isolated pixel occurring in class 'object' is very low. In this situation the $P(x|y)$ cannot be maximised for each pixel independently and instead an attempt is made to maximise their product over all pixels. This is equivalent to minimising the sum over all pixels, of the exponent from Eq B-4, often termed the Gibbs energy function:

$$U(x \mid y) = \sum_i \{ \ln(\sigma_{x_i}) + \frac{(y_i - \mu_{x_i})^2}{2\sigma_{x_i}^2} + \sum_{n=1}^{N} [\theta_n J(x_i x_{i+n})] \} \qquad \text{Eq B-5}$$

where the $P(x)$ are now modelled by a Markov random field in which for each pixel, i, a sum is made over a defined neighbourhood set of N pixels. The function $J(x_i, x_{i+n})$ returns a value of -1 whenever the label for the n^{th} neighbourhood pixel x_{i+n} is the same as the label for the central pixel i and otherwise is zero. The terms θ_n can be adjusted to vary the significance of different neighbours as well as the relative weighting between the $P(x)$ and the $P(y|x)$ components.

The optimal segmentation of the image will therefore be achieved by minimising the energy function, Eq B-5, by appropriate choice of classes for all pixels. This equation has as many variables as there are pixels in the image and cannot therefore be exhaustively analysed to locate the minimum except for trivial applications. Various optimisation methods are available. Here, simulated annealing, a powerful but relatively slow technique has been applied and a detailed description of this method has already been given in chapter 6.

A preliminary classification is needed to start the optimisation and typically pixels are placed in those classes for which the class-conditional probabilities are a maximum thereby ignoring the dependency on neighbouring pixel classes which of course initially are unassigned.

B.1 Parameter Estimation

To apply this technique to image segmentation it is necessary first to determine the means, μ_x, and variances, σ_x^2, for each class label x and the neighbourhood consistency parameters θ_n.

The means and variances can be found by manually segmenting a representative image, deciding which regions are in which class and then calculating the statistics for each class. Alternatively using the total image histogram, Gaussian curves can either be fitted analytically by defining a goodness of fit measure which is then minimised by altering the Gaussian parameters or they can be simply sketched in by hand to yield adequate parameter estimates.

The parameter θ_n represents the weighting which should be given to the local agreement between pixel labels. For example, a pixel which is presently labelled as 'object' should have a higher probability of being correct if more of its neighbouring pixels were identically labelled. The analytical approach to determining θ_n would involve initially labelling all pixels according to pixel attribute alone (*i.e.* in the absence of any contextual information). Every labelled pixel in the image would then be examined and a note would be made of the number of identically labelled neighbouring pixels. Counts would be made for every combination of pixel label and number of agreeing neighbouring pixel labels. These would be used to infer how the probability of the central pixel label being correct varies with the number of consistent neighbouring labels which would in turn be used to generate estimates for θ_n . In practice, most implementations treat θ_n as a constant θ which is initially estimated at around unity and adjusted until optimal results are obtained. A higher value of θ will emphasise local agreement of pixel labels over the class suitability according to pixel attribute and *vice versa*.

References:

---

[1] R. Nevatia, Image Segmentation, Chapter 9, Handbook of Pattern Recognition and Image Processing, ed; T.Y. Young and K-S Fu, Academic Press.

[2] W.K. Pratt, Digital Image Processing (second edition) Wiley, New York.

[3] K. Appel and W. Haken, Every planar map is four colorable. Part I. Discharging, Illinois J. Math., 21, 429-490, 1977.

[4] S. Wang and R.M. Haralick, Automatic Multithreshold Selection, Comp. Vis. Gra. Im. Proc., 25, 46-67, 1984.

[5] I. Hannah, D. Patel and R. Davies, The Use of Variance and Entropic Thresholding Methods for Image Segmentation, Patt. Rec., 28, 1135-1143, 1995.

[6] P. Courtney, N. Thacker and A. Clarke, Algorithmic Modeling for Performance Evaluation, Machine Vis. and Apps., 9, 219-228, 1997`

[7] X. Yu, T.D. Bui and A. Krzyzak, Robust Estimation for Range Image Segmentation and Reconstruction, IEEE Trans. PAMI, 16, 530-537, 1994.

[8] B. Bhanu, S. Lee and S. Das, Adaptive Image Segmentation Using Genetic and Hybrid Search Schemes, IEEE Trans. Aero. and Elec. Sys., 31, 1268-1290, 1995.

[9] R.M. Haralick, Digital Step Edges from Zero Crossing of Second Directional Derivatives, IEEE Trans. PAMI, 6, 58-68, 1984.

[10] L.D. Griffin, A.C.F. Colchester and G.P. Robinson, Scale and Segmentation of grey-level images using maximum gradient paths, Image and Vis. Comp., 10, 389-402, 1992.

[11] M. Tabb and N. Ahuja, Multiscale Image Segmentation by Integrated Edge and Region Detection, IEEE Trans. Im. Proc., 6, 642-655,1997.

[12] A. Khotanzad and A. Bouarfa, Image Segmentation by a Parallel, Non-Parametric Histogram Based Clustering Algorithm, Patt. Rec., 23, 961-973, 1990.

[13] R. Adams and L. Bischof, Seeded Region Growing, IEEE Trans. PAMI, 16, 641-647, 1994.

[14] M. Spann and C. Horne, Image Segmentation Using a Dynamic Thresholding Pyramid, Patt. Rec., 22, 719-732, 1989.

[15] S.S. Sinja and B.G. Schunck, A Two-Stage Algorithm for Discontinuity-Preserving Surface Reconstruction, IEEE Trans. PAMI, 14, 36-55, 1992.

[16] E. Trucco and R.B. Fisher, Experiments in Curvature-Based Segmentation of Range Data, IEEE Trans. PAMI, 17, 177-182, 1995.

[17] T.C. Pong et al, Experiments in Segmentation using a Facet Model Region Grower, Comp.Vis. Gra. Im. Proc., 25, 1-23, 1984.

[18] P.J. Besl and R.C. Jain, Segmentation Through Variable-Order Surface Fitting, IEEE Trans. PAMI, 10, 167-192, 1988.

[19] T. Hofmann, J. Puzicha and J.M. Buhmann, Unsupervised Texture Segmentation in a Deterministic Annealing Framework, IEEE Trans PAMI, 20, 803-818, 1998.

[20] P.P. Raghu and B. Yegnanarayana, Segmentation of Gabor-Filtered Textures Using Deterministic Relaxation, IEEE Trans. Im. Proc., 5, 1625-1635, 1996.

[21] O. Pichler, A. Teuner and B.J. Hosticka, An Unsupervised Texture Segmentation Algorithm with Feature Space Reduction and Knowledge Feedback, IEEE Trans. Im. Proc., 7, 53-61, 1998.

[22] U. Bhattacharya, B.B. Chaudhuri and S.K. Parui, An MLP-based texture segmentation method without selecting a feature set, Image and Vis. Comp., 15, 937-948, 1997.

[23] L. Gupta and T. Sortrakul, A Gaussian-Mixture-Based Image Segmentation Algorithm, Patt. Rec., 31, 315-325, 1998.

[24] P. Thevenaz, U.E. Ruttimann and M. Unser, A Pyramid Approach to Subpixel Registration Based on Intensity, IEEE Trans. Im. Proc., 7, 27-41, 1998.

[25] C. Bouman and B. Liu, Multiple Resolution Segmentation of Textured Images, IEEE Trans. PAMI, 13, 99-113, 1991.

[26] D. Dunn and W.E. Higgins, Optimal Gabor Filters for Texture Segmentation, IEEE Trans. Im. Proc., 4, 947-964, 1995.

[27] A. Laine and J. Fan, Texture Classification by Wavelet Packet Signatures, IEEE Trans. PAMI, 15, 1186-1191, 1993.

[28] R. Porter and N. Canagarajah, A Robust Automatic Clustering Scheme for Image Segmentation Using Wavelets, IEEE Trans. Im. Proc., 5, 662-665, 1996.

[29] G.L. Gimelfarb, Texture Modelling by Multiple Pairwise Pixel Interactions, IEEE Trans. PAMI, 18, 1110-1114, 1996.

[30] V.P. Kumar and U.B. Desai, Image Interpretation Using Bayesian Networks, IEEE Trans. PAMI, 18, 74-76, 1996.

[31] H. Derin and H. Elliott, Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields, IEEE Trans. PAMI, 9, 39-55, 1987.

[32] R.C. Dubes et al, MRF Model-Based Algorithms for Image Segmentation, Proc 10th Int'l Conf. Patt. Rec., 808-814, 1990.

[33] I.Y. Kim and H.S. Yang, An Integration Scheme for Image Segmentation and Labelling Based on Markov Random Field Model, IEEE Trans. PAMI, 18, 69-73, 1996.

[34] S. Geman and D. Geman, Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images, IEEE Trans. PAMI, 6, 721-741, 1984.

[35] J. Besag, On The Statistical Analysis of Dirty Pictures, Jou. Roy. Statist. Soc. B., 48, 259-302, 1986.

[36] S. Sarkar and K.L. Boyer, Perceptual Organisation in Computer Vision: A Review and a Proposal for a Classificatory Structure, IEEE Trans. Sys, Man and Cyb., 23, 382-398, 1993.

[37] S. Liou, A.H. Chiu and R.C. Jain, A Parallel Technique for Signal-Level Perceptual Organisation, IEEE Trans. PAMI, 13, 317-325, 1991.

[38] P. Andrey and P. Tarroux, Unsupervised Image Segmentation using a Distributed Genetic Algorithm, Patt. Rec., 27, 659-673, 1994.

[39] P.P Raghu, R. Poongodi and B. Yegnanarayana, Unsupervised Texture Classification Using Vector Quantisation and Deterministic Relaxation Neural Network, IEEE Im. Proc., 6, 1376-1387, 1997.

[40] J. Le Moigne and J.C. Tilton, Refining Image Segmentation by Integration of Edge and Region Data, IEEE Trans. Geo. and Rem. Sens., 33, 605-615, 1995.

[41] A. A. Farag and E.J. Delp, Edge Linking by Sequential Search, Patt. Rec., 28, 611-633, 1995.

[42] S.M. LaValle and S.A. Hutchinson, A Bayesian Methodology for Parametric Image Models, IEEE Trans. PAMI, 17, 211-217, 1995.

[43] T. Kohonen et al, Engineering Applications of the Self-Organising Map, IEEE Proc. 84, 1358-1383, 1996.

[44] C.V. Stewart, Bias in Robust Estimation Caused by Discontinuities and Multiple Structures, IEEE Trans. PAMI, 19, 819-833, 1997.

[45] C. A. Bouman and M. Shapiro, A Multiscale Random Field Model for Bayesian Image Segmentation, IEEE Trans. Im.. Proc., 3, 162-177, 1994.

[46] S. Ghosal and R. Mehrotra, Segmentation of Ranges Images: An Orthogonal Moment-Base Integrated Approach, IEEE Trans. Rob. Aut., 9, 385-399, 1993.

[47] Y. Delignon, A. Marzouki and W. Pieczynski, Estimation of Generalised Mixtures and its Application in Image Segmentation, IEEE Trans. Im. Proc., 6, 1364-1381, 1997.

[48] J-M. Beaulieu and M. Goldberg, Hierarchy in Picture Segmentation: A Stepwise Optimisation Approach, IEEE Trans. PAMI, 11, 150-163, 1989.

[49] R.M. Haralick and L. Watson, A facet model for image data, Comp. Gra. Im. Proc. 15, 113-129, 1981.

[50] K.S. Fu and K.K. Mui, A Survey on Image Segmentation, Patt. Rec., 13, 3-16, 1980.

[51] N.R. Pal and S.K. Pal, A Review of Image Segmentation Techniques, Patt. Rec., 26, 1277-1294, 1993.

[52] Y.J. Zhang, A survey on Evaluation Methods for Image Segmentation, Patt. Rec., 29, 1335-1346, 1996.

[53] R.M. Haralick and L.M. Shapiro, Image Segmentation Techniques, Comp. Vis. Graphics and Im. Proc., 29, 100-132, 1985.

[54] P.K. Sahoo, S. Soltani and A.K. Wong, A Survey of Thresholding Techniques, Comp. Vis. Graphics and Im. Proc., 41, 233-260, 1988.

[55] D. Marr and E. Hildreth, Theory of Edge Detection, Proc. Roy. Soc. Lond., B 207, 187-217, 1980.

[56] A.D. Brink, Minimum Spatial Entropy Threshold Selection, IEE Proc. Vis. Im. Sig., 142, 128-132, 1995.

[57] Chion and Hwang, A Neural Network-Based Stochastic Active Contour Model (Snake) for Contour Finding of Distinct Features, IEEE Trans. Im. Proc., 1407-1423, 1995.

[58] A.P. Choo, A.J. Maeder and B. Pham, Image Segmentation for Complex Natural Scenes, Image and Vis. Comp., 8, 155-163, 1990.

[59] A. Deruyver and Y. Hode, Automatic Multi-Thresholdable Image Segmentation using Separating Bipoints, Proc. 13th ICPR, B, 457-461, 1996.

[60] T. Gevers and V.K. Kajcovski, Image Segmentation by Directed Region Division, Proc. 12th ICPR, A, 342-346, 1994

[61] L.D. Griffin,et al, Hierarchical Segmentation Satisfying Constraints, Proc. 5th BMVC Conf, 1, 135-144, 1994

[62] J.F. Haddon and J.F. Boyce, Image Segmentation by Unifying Region and Boundary Information, IEEE Trans. PAMI, 12, 929-948, 1990.

[63] J. Ivins and J. Porrill, Statistical Snakes: Active Region Models, Proc. 5th BMVC Conf., 2, 377-386, 1994

[64] I. Koch and G. Marshall, Bootstrap Plots for Image Segmentation, Proc. 13th ICPR, B, 447-451, 1996

[65] X.Q. Li,et al, Fuzzy logic Approach to Image Segmentation, Proc. 12th ICPR, A, 337- 341,1994

[66] M. Lybanon, S.M. Lea and S.M. Hines, Segmentation of Diverse Image Types Using Opening and Closing, Proc. 12th ICPR, A, 347-351, 1994.

[67] M.C. Morrone A. Navarngione and D. Burr, An Adaptive Approach to Scale Selection for Line and Edge Detection, Patt. Rec. Lett., 16, 667-677, 1995.

[68] S.S. Iyengar and D. Weian, An Efficient Edge-Detection Algorithm Using Relaxation Labelling Technique, Patt. Rec., 28, 519-536, 1995.

[69] W. Snyder et al, Segmentation of Magnetic Resonance Images using Mean Field Annealing, Image and Vis. Comp., 10, 361-368, 1992.

[70] A. Toet, Hierarchical Clustering Through Morphological Transformation, Patt. Rec. Lett., 12, 391-399, 1991.

[71] D. Tsai, A Fast Thresholding Selection Procedure for Multimodal and Unimodal Histograms, Patt. Rec. Lett., 16, 653-666, 1995.

[72] J. Teuber, Digital Image Processing, Prentice Hall, 1993

[73] K.R. Castleman, Digital Image Processing, Prentice Hall, 1996

[74] J. Canny, A Computational Approach to Edge Detection, IEEE Trans. PAMI, 8, 679-697, 1986.

[75] J.D. Browning and S.L. Tanimoto, Segmentation of pictures into regions with a tile by tile method, Patt. Rec., 15, 1-10, 1982.

[76] J.T. Tou and R.C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, 1974

[77] A.P. Witkin and J. Tenenbaum, The role of structure in vision, Human and Machine Vision, Beck et al, eds, New York Academic, 1983.

[78] T.R. Reed and J.M. Hans Du Buf, A Review of Recent Texture Segmentation and Feature Extraction Techniques, Comp. Vis. Gra. Im. Proc., 57, 359-372, 1993.

[79] J.Y. Hsiao and A.A. Sawchuk, Supervised Textured Image Segmentation Using Feature Smoothing and Probabilistic Relaxation Techniques, IEEE Trans. PAMI, 11, 1279-1292, 1989.

[80] R.M. Haralick, Statistical Image Texture Analysis, Chapter 11, Handbook of Pattern Recognition and Image Processing, ed; T.Y. Young, and K-S Fu, Academic Press, 1985.

[81] J.C. Russ, The Image Processing Handbook, CRC Press, 1994.

[82] R.M. Haralick et al, Textural Features for Image Classification, IEEE Trans. Sys. Man and Cyb., 3, 610-621,1973.

[83] G. Schafer, A Mathematical Theory of Evidence, Princeton University Press, 1976.

[84] R. Horst and P.M. Pardalos (eds.), Handbook of Global Optimization, Kluwer, Dordrecht, 1995.

[85] J.D. Pinter, Optima, the Newsletter of the Mathematical Programming Society, 52, 1-8. 1996.

[86] N. Metropolis et al, Equations of state calculations by fast computing machines, Jou. Chem. Phys., 21, 1087-1091,1953.

[87] K. Konstantinides and J.R.Rasure, The Khoros Software Development Environment for Image and Signal Processing, IEEE Trans. Im. Proc., 3, 243-252, 1994.