UNIVERSITY OF SOUTHAMPTON

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS

Institute of Sound and Vibration Research

Connected Operators for Unsupervised Image Segmentation

by

Oliver Nicholas Baumann

Thesis for the degree of Doctor of Philosophy

December 2004

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS

INSTITUTE OF SOUND AND VIBRATION RESEARCH

Doctor of Philosophy

CONNECTED OPERATORS FOR UNSUPERVISED IMAGE SEGMENTATION

by Oliver Nicholas Baumann

Image segmentation forms the first stage in many image analysis procedures including image sequence re-timing and the emerging field of content based retrieval. By dividing the image into a set of disjoint connected regions, each of which is homogeneous with respect to some measure of the image content, the scene can be analysed and metadata extracted more efficiently, and in many cases more effectively, than on a pixel by pixel basis. Though a great number of segmentation techniques exist (and continue to be developed,) many of them fall short of the requirements of these applications. This thesis first defines these requirements and reviews established segmentation methods describing their qualities and shortfalls. Selecting the watershed transform and connected operators from those techniques reviewed a number of novel adaptations are introduced, developed and shown to produce pleasing results both in terms of a new evaluation metric and subjective appraisal. Finally, the use of the image segmentation is shown to improve established methods of image noise removal using the discrete wavelet transform.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my gratitude firstly to Paul White for his supervision and guidance in conducting this research and to Snell & Wilcox for their sponsorship and in particular Mike Knee, Martin Weston and Kaaren May for their counsel. Gratitude also to Bill Collis and Antonello De Stefano for their advice and technical assistance. Finally, thanks are due to Sarah, Gretel, Dan and of course my Mum and Dad for their continuing support throughout the last few years.

For Ivy

Chapter 1

Introduction

1.1 Context and Motivation

In the last 30 years the field of image processing has received considerable attention. In particular the use of digital computers for the representation, processing and analysis of images has become a vast and very popular field of signal processing with applications including automatic inspection, medical image analysis and object or pattern recognition. These applications are high-level in that they attempt to derive semantic knowledge of the image content, a problem which given a matrix of pixels is not trivial. As such, many of these high-level applications require processing techniques which simplify image content by grouping pixels into disjoint regions which are meaningful with respect to image content. These processing techniques are broadly referred to as image segmentation.

Two basic types of segmentation exist; *complete*, in which the disjoint regions of the segmentation correspond directly with objects in the image, and *partial*, where regions are defined as being homogeneous with respect to one or more properties such as grey level, colour, shape or motion. Partial segmentations are of most interest to this study because they are highly versatile and require little or no prior knowledge of the image content. They are used extensively as a pre-processing stage to higher level image processing tasks including object recognition, image coding and in the case of image sequences, the estimation of motion vectors and re-timing. It may seem paradoxical that motion vectors can be both a criterion for, and an application of image segmentation but in fact segmentation and motion estimation are interlinked problems, in that a good motion estimate can help greatly in obtaining a good segmentation while a good segmentation will undoubtedly help in producing a good motion field.

A considerable number of established techniques exist for the partial segmentation of still images, most of which fall into three main groups referred to as global, edge and region based methods. Global segmentation methods as the name suggests use only global knowledge of the image to separate it into a number of regions. This knowledge usually takes the form of a histogram and a number of methods exist for separating the image into regions based on the different modes thereof. Such methods take no account of local information and as such often result in a relatively small number of regions which are not necessarily spatially connected. In this thesis we constrain our attention to segmentations which produce connected regions.

Edge based methods seek to define regions of the segmentation by resolving their boundaries. The first stage usually involves the application of a low-level edge detector such as the Sobel or Prewitt operators which highlight discontinuities in the image producing an estimate of first derivative or the Laplacian operator which estimates the second derivative of the image function. The first- or second-order gradient image is then subjected to some form of thresholding or zero-crossing detection the result of which, it is hoped, is a set of closed contours the interiors of which are homogeneous with respect to grey level. In fact these methods are highly susceptible to noise and frequently miss edges of distinct homogeneous regions and/or create contours where no region boundaries exist. More successful edge based methods such as the Hough transform attempt a complete segmentation and require a considerable amount of prior knowledge about the objects to be segmented and/or user interaction. This reduces the generality of the method which for this research is undesirable.

It is region based methods which form the most significant line of investigation of this research. Rather than defining segments by their boundaries, region based methods define the segments directly. There are two basic philosophies for the generation of region based segmentations, splitting and merging, which can be used alone or in unison. Splitting algorithms start with one region representing the entire image which is split into quadrants whose homogeneity with respect to the chosen property is assessed. A decision is made to split the image if the homogeneity of each region is improved. The process is repeated for each new region until the homogeneity of each region is within the desired limits. The use of quadrants, while convenient, places undesirable restrictions on the shape of regions. One further drawback is that the splitting of regions relies on a global assessment of the homogeneity which ignores local information and can lead to ambiguous region definitions.

Split-and-merge approaches represent the image as a quad-tree pyramid structure allowing image quadrants to be both split in four, as with splitting techniques, and merged with neighbouring regions if the homogeneity of the union is improved. Though this partially overcomes the global criterion assessment problem associated with splitting, the quad-tree structure still restricts region shape.

Merging algorithms begin with an initial segmentation which will tend to have a very large number of regions and may even define each pixel of the image as a distinct region. Traditionally, regions are then merged such that the homogeneity criterion is maintained or improved. One popular method for producing the initial segmentation is the watershed transform for which homogeneity requires that every member pixel belongs to the same catchment basin in that it is connected to the regional minimum of that segment by a path in which the grey level monotonically decreases. It is common for the watershed transform to be applied to the gradient image providing a segmentation image whose segments have natural boundaries which in many cases correspond to edges in the image. In fact, the watershed algorithm has been used very extensively to obtain partial segmentations and a section of this thesis is devoted to the development of schemes which aim to improve these results. Apart from being intuitively suitable for partial segmentation, region based segmentation methods are inherently robust in the presence of noise. It is for this reason that these methods form the basis of study for this research.

The set of operators referred to as connected operators are of particular interest as in two dimensions they have been shown to be remarkably versatile. Connected operators are initiated by a partition of the image which can be based upon any criterion such as flat grey level or some measure of colour space. The operator then merges these regions based upon the measurement of some property of each region. For example, the initial partition may consist of regions which are flat with respect to grey level, the operator may require that if a region is both a regional maximum and smaller than a given threshold then that region is given the grey level of the highest neighbour. Such a filter is the morphological operator of *area opening.* The set of morphological operators are in fact a subset of connected operators. They affect the image depending on local shape structures and are capable of removing image details such as texture information whilst maintaining the basic contour and shape information.

This research investigates the application of a number of existing connected operators to image segmentation. A metric for the evaluation of the resulting segmentations is proposed and novel post-processing connected operators, which are shown to improve results are developed. Having shown the segmentation evaluation metric to be useful in selecting a suitable segmentation form a number of possibilities, a connected operator is specifically designed to minimise the metric measure with mixed success. Further, the application of connected operators at various stages of the watershed technique are shown to give improvements over traditional approaches. Finally, a spatially adaptive wavelet based de-noising strategy using the segmentation techniques developed is proposed and shown to offer significant improvements over global methods.

The contributions made by this research are as follows.

- A metric for the evaluation of image segmentations is proposed and developed to allow control of the degree of segmentation to suit various images and applications.

- The use of connected operators at various stages of the watershed technique is shown to reduce its tendency to over-segment.

- A novel operator which produces a segmentation by minimising our evaluation metric is proposed.

- Segmentation of the kind proposed in this thesis is applied to the method of wavelet noise reduction and shown to improve the increase in output signal to noise ratio over the un-segmented noise reduction scheme.
- A post-processing connected operator for the improvement of segmentations resulting from existing connected operators is developed.
- A study of the use of connected and morphological operators for the removal of impulsive noise from grey scale images is presented.

1.2 Layout of the Thesis

This thesis begins with a review of the techniques available for the segmentation of still images including global, edge, region and morphological techniques. Of these, the region based techniques are deemed most suitable for development and more specifically connected operators and the watershed algorithm. As such Chapter 3 introduces a unifying framework, that of the lattice of partitions, and defines these image operators in the context thereof.

In Chapter 4 we first introduce an objective segmentation evaluation metric and demonstrate that connected operators can be used for the segmentation of images but for the issue of localised over-segmentation. This is addressed with the introduction of novel operators for its mitigation which are shown to improve results both in terms of the new metric and subjective appraisal. The final section of this chapter introduces a novel connected operator the aim of which is to minimise the evaluation metric.

Chapter 5 introduces two novel methods by which the tendency of the watershed transform to over-segment can be indirectly controlled. Of these the method of gradient smoothing is shown to produce segmentations which meet our requirements. This method is then demonstrated to increase the performance of established noise reduction techniques which use the wavelet decomposition.

Finally, the conclusions of our research and suggestions for further work are presented in Chapter 6.

Chapter 2

A Review of Partial Segmentation Techniques

Segmentation is the splitting of an image into regions or segments each of which has some significance with regard to the image function. The most basic requirements of a segmentation are that the segments R_i are disjoint, in that they do not overlap, and that the union of the segments represents the entire domain of the image function under study D. These requirements are summarised in the following definition:

Definition 1 (Partition of a Space (Segmentation)) A partition P of a space, D, is a set of connected components or partition classes, $\{R_i\}$, which are disjoint, $R_i \cap R_j = \emptyset \quad i \neq j$, and the union of which is the entire space, $\bigcup_i R_i = D$.

Regions of a complete segmentation are required to correlate with objects, as humans perceive them. For partial segmentation, though a reasonable correlation with objects may be hoped for, segments are required only to be homogeneous with respect to some property. This homogeneity is a fairly loose concept and not necessarily just grey level 'flatness'. Complete segmentation often requires semantic knowledge of the scene content making techniques for their production highly application specific. Our attentions concentrate on techniques for the partial segmentation of images since they usually require no semantic knowledge and are therefore very versatile.

This chapter introduces the main methods of partial image segmentation of still images documented in the literature. In order to give direction to our study, we must outline some properties which we believe the resulting segmentation images must have. The first requirement is that each segment consists of a set of connected pixels. Though the notion of a connected set is open to some interpretation, this criterion imposes the constraint that a single segment should not represent more than one remote region. The second criterion relates to the boundaries of the segments. Though it may be reasonable for a single object in the image to be represented by several segments, it is usually less desirable to have several objects in the same segment. As such we require that the edges of segments correspond well with object boundaries. We also wish for the number of segments to be considerably less than the number of pixels in the image for any improvement over pixel level motion estimation to be significant. One further restriction we place on the technique regards the information which is made available to it. The method of segmentation must use only the information in a grey level image or intensity function i.e. no knowledge of object in the scene or user interaction. We restrict our study to the use of grey level information for segmentation, though the techniques may be extended to take advantage of other colour spaces with more than one dimension. And finally we require that the process be as computationally efficient as possible. These requirements are summarised below.

- The information required for the segmentation must be derived from the image intensity (grey scale) function.
- The technique must require no semantic knowledge of the scene or user interaction.
- Segments must comprise connected sets of pixels.
- Segment boundaries should correspond well with object boundaries.
- Computational load must be kept to a minimum.

The conditions given above are required of segmentations used in many applications including but not limited to the estimation of motion vectors, object recognition and content based retrieval (as permitted by the MPEG-7 standard), the latter two being relatively new fields of study. A discussion of more generic criteria for a segmentation are discussed in the context of the segmentation evaluation metric in Chapter 4.

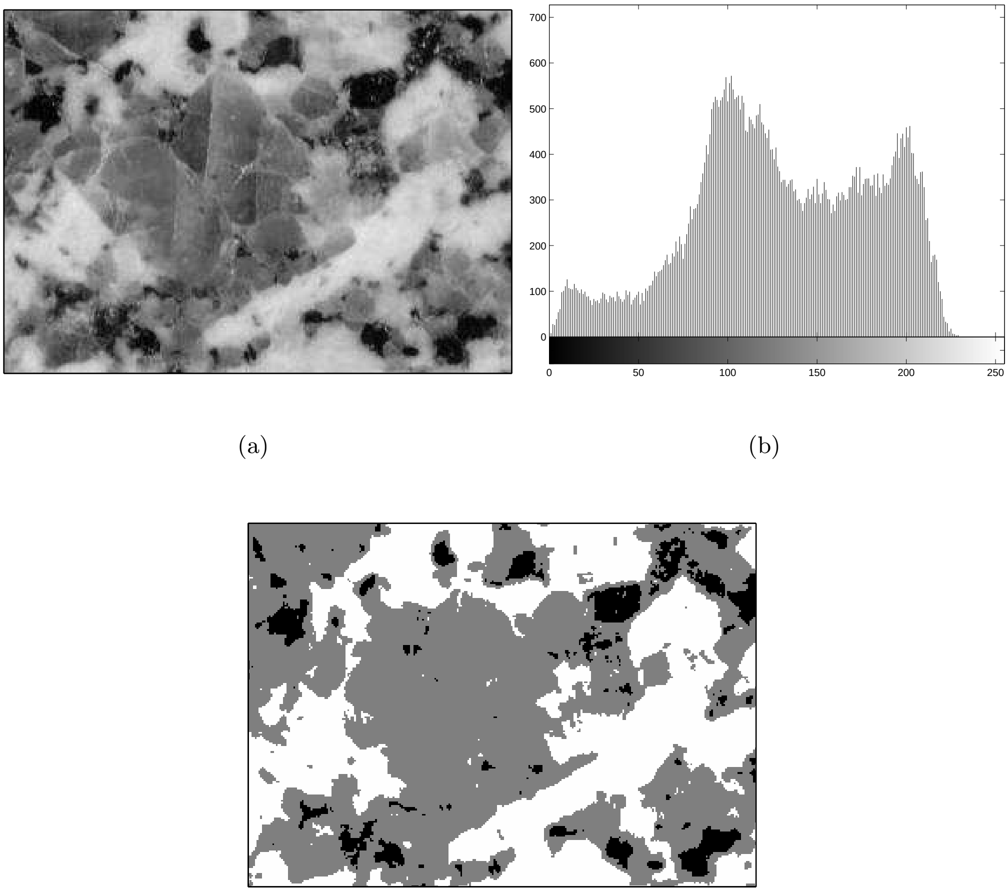
Partial segmentation techniques fall into three main categories, global, edge and region based methods, to each of which a section of this chapter is devoted, covering the most established algorithms in the field. The final section of this chapter details the widely used morphological technique of the watershed transform which is neither entirely edge nor entirely region based.

2.1 Global Segmentation Methods

Global segmentation methods use only global information about the image, this information usually takes the form of the intensity or grey level histogram but the histogram may equally be based upon the gradient (see section 2.2) or texture [1] of the image function. Using the grey level histogram, the image is split into two or more regions the comprising pixels of which all have a similar grey level. Figure 2.1(a) shows an image of a mineral sample having three distinct phases appearing as black, grey and light grey to white. The grey level histogram, given in Figure 2.1(b), displays three significant modes lying in the approximate ranges; 0-50, 51-150, and 151-255. These modes correspond with the three phases of the sample and as such, splitting the image into three regions each consisting of pixels having grey levels in the three ranges results the image presented in Figure 2.1(c). Such a segmentation is valid in that it conforms with Definition 1 and in this application is of some use, however the regions which are defined are not necessarily connected.

In the example given above, the band thresholds were chosen by visual inspection of the histogram but methods exist for the automatic determination of band thresholds which include searching for minima [2], and modelling the histogram as a mixture of Gaussian distributions [1]. Whilst these methods remove the need for interaction, they do not solve the problem of disconnected segments.

It is clear that global segmentation techniques are unsuccessful in meeting our requirements. As such, we now concentrate on methods which exploit local knowledge

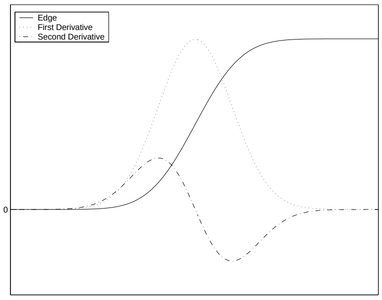


(a)

(b)

(c)

Figure 2.1: The original image of a section of polished granite (a), the histogram thereof (b), and the result of thresholding the image based upon the three dominant modes (c).



(a)

Figure 2.2: A synthetic one-dimensional edge and the first and second derivatives thereof.

in the image.

2.2 Edge Based Segmentation Methods

The first step in any edge based segmentation is edge detection. Though methods for the detection of edges in colour images have been proposed [1, 3], the most established methods involve highlighting rapid changes in the image intensity function (grey level) in order to produce a binary *edge image* with pixel values corresponding to the states 'edge' or 'not edge'. It is to the methods of edge detection that we restrict the discussion of this section.

The majority of edge detection methods assume a step edge model in which edges of objects are assumed to correspond with rapid, high contrast changes in grey level which in turn correspond to features in the first- and second-order gradient of the image. Figure 2.2 shows a one dimensional discontinuity along with its first and second order gradient functions. Though the levels of the derivatives have been accentuated for illustrative purposes, it is clear that the location of the discontinuity is signified by the peak in the first derivative and the zero-crossing in the second. The gradient of a continuous function in two-dimensions $f(x, y)$ is given by equation 2.1 and the second derivative, frequently referred to as the Laplacian, by equation

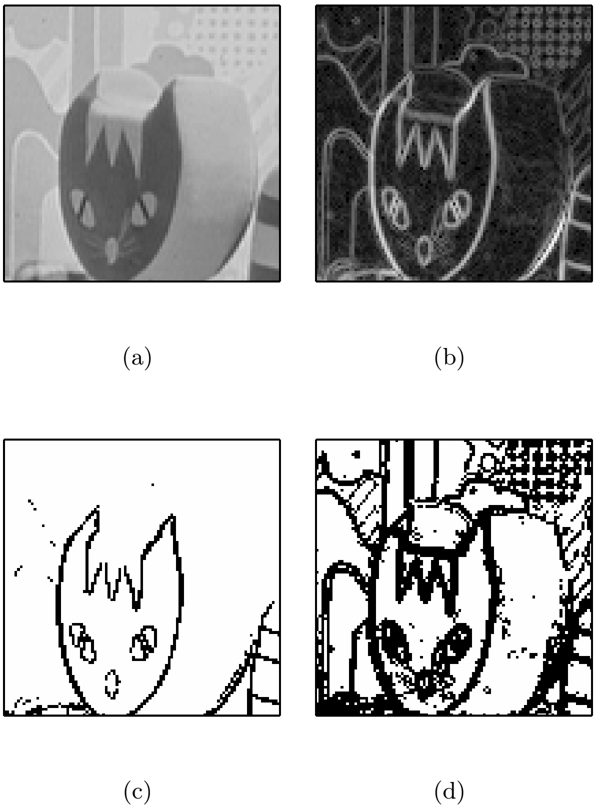
2.2.

$$|\nabla f(x,y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \tag{2.1}$$

$$\nabla^2 f(x,y) = \left(\frac{\partial^2 f}{\partial x^2}\right) + \left(\frac{\partial^2 f}{\partial y^2}\right) \tag{2.2}$$

However, the partial derivatives of discrete data cannot be obtained directly, and so must be approximated by differences.

Several methods exist for the estimation of the first derivative of a digital image, all of which rely on the convolution of the signal with a number of kernels each responding more strongly to a gradient in one specific direction such kernel sets include the Sobel and Prewitt operators [1]. The gradient magnitude is most commonly evaluated using the Euclidean norm of the results of all the convolutions and the gradient direction by the kernel responding most strongly or some trigonometric calculation based on the contribution of orthogonal kernels. Where edges exist, their direction can be arranged to lie perpendicular to the gradient direction and it can therefore be useful in edge linking. Having estimated the gradient, the image is subjected to a thresholding in which pixels having an associated gradient higher than a specified value are given the label 'edge' and those below the value 'not edge'. It is the determination of this threshold value that poses most difficulty. Figure 2.3(a) shows a section of the original image of 'girl', the gradient magnitude of which is calculated by taking the Euclidean norm of the convolutions of the image with the vectors $[1, 0, -1]$ and $[1, 0, -1]^T$, Figure 2.3(b). This image is subject to thresholding at levels equal to one quarter and one twentieth of the maximum value, the results of which are shown in Figures 2.3(c) and 2.3(d) in which the 'edge' state is shown in black. It is clear from the results that the higher of the two thresholds misses important edges including those of the back of the head of the toy and although the lower threshold detects more of these edges in the image function, it has also detected a number of discontinuities which do not correspond to object boundaries and the thickness of those edges detected for a higher threshold has increased. Thick edges are undesirable because pixels defined as edges are not part of any region and in order to extract a tessellation as required by Definition 1 each 'edge' pixel must be assigned a region. Edges greater than a single pixel in width leave some ambiguity as to the exact location of the edge and so correlation of edges with object boundaries is compromised. Thin edges on the other hand being of single pixel thickness can



(a) (b)

(c) (d)

Figure 2.3: A section of the original image of ‘girl’ (a), the normalised gradient thereof (b) and examples of thresholding the gradient image at one quarter (c) and one twentieth (d) of its range.

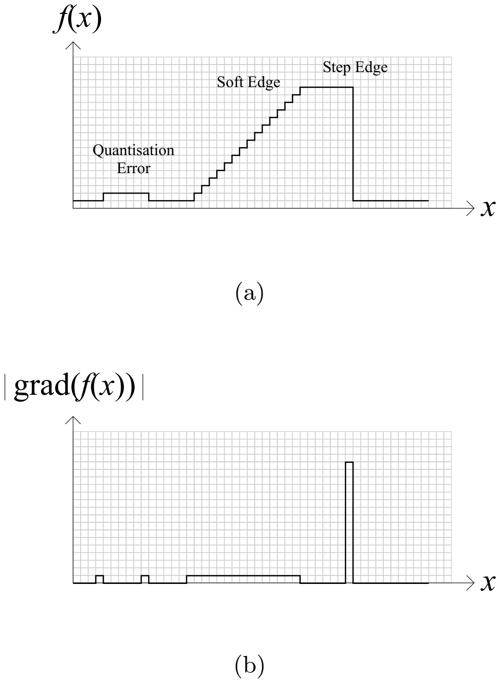


Figure 2.4: A synthetic one-dimensional signal (a) and the gradient thereof (b).

easily be assigned to a region without loosing edge accuracy. Only in very few cases, where objects are very distinct from the background but are otherwise homogeneous with respect to grey level does the result of gradient thresholding give results which can be used directly to form a segmentation. This is due to the fact that edge detection strategies requiring the estimation of the gradient are highly sensitive to quantisation error and noise whilst being relatively insensitive to soft edges. Figure 2.4(a) shows a synthetic one dimensional signal with three features; quantisation error or noise, a soft edge or ramp and a sharp or step edge. It should be noted that the latter is highly unlikely to occur in real images due to the finite sampling grid and anti-aliasing requirements. Figure 2.4(b) shows the gradient of the signal calculated by convolution with the kernel $[-1, 1]$. Whilst the sharp edge gives rise to a large peak in the gradient function, the soft edge gives no higher a response than the quantisation noise. Thresholding the gradient of this function will result in either the detection of all three features or only the step edge. While methods such as non-maximal suppression and hysteretic thresholding [1] can improve the standard gradient thresholding techniques, reducing their response to noise, thinning the regions of state ‘edge’ and increasing the correlation between detected edges and object boundaries, the resulting images still suffer from the problem that the lines defined as edges do not form closed contours from which a tessellation of regions can be easily extracted. Even more computationally expensive methods such as edge relaxation when applied to gradient thresholding results suffer from this problem. The drawbacks of first order gradient thresholding including thick edges, and a tendency to produce open contours are overcome to some extent by methods which use the second order gradient or Laplacian of the image.

It has been demonstrated that the second derivative of an image function has a zero-crossing at every extremum of the first derivative (Figure 2.2). As such, detecting the zero-crossings in the Laplacian of an image function will result in huge over-detection, particularly in the presence of noise or quantisation error. One method of reducing the sensitivity is to smooth the image using a linear filter prior to estimating the second derivative. The method discussed here is due to Marr and Hildreth [4]. The filter of choice is that of the linear Gaussian filter, being the optimal compromise between spatial support and frequency bandwidth. The effect of convolving the Gaussian kernel with the image function is to remove noise and soften edges. The second derivative or Laplacian of the smoothed image is then estimated. An expression for the complete process is given in Equation 2.3.

$$\nabla^2 \left[G(x,y,\sigma) * f(x,y) \right] \tag{2.3}$$

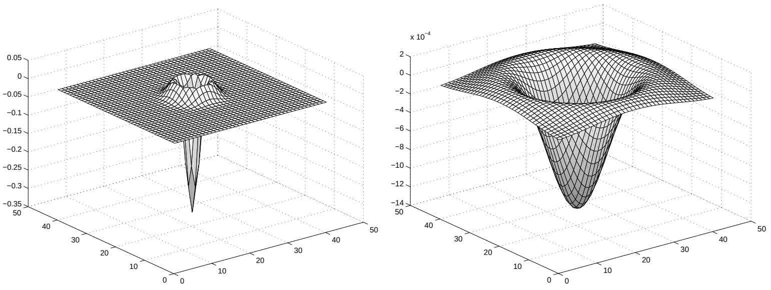
But, since all the operators are linear and therefore associative, we can change the order of convolution and the Laplacian operator such that we have:

$$\left[\nabla^2 G(x,y,\sigma) \right] * f(x,y) \tag{2.4}$$

This considerably reduces the complexity of the operation as the Laplacian of Gaussian (LoG) can be computed analytically [1] to be:

$$\mathrm{LoG}(x,y,\sigma) = \left[\nabla^2 G(x,y,\sigma) \right] = \frac{1}{\pi\sigma^4} \left[2 - \frac{x^2+y^2}{2\sigma^2} \right] e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.5}$$

Examples of the LoG kernel are given in Figure 2.5 for values of σ equal to 1 and 4. Figure 2.6(a) shows the original image of girl along with the results of the zero-crossing detection after convolution with a LoG kernel with σ equal to 2, 4 and 6. The Laplacian of Gaussian method of edge detection appears to have some very desirable properties, the lines in the edge image are thin and though no constraint

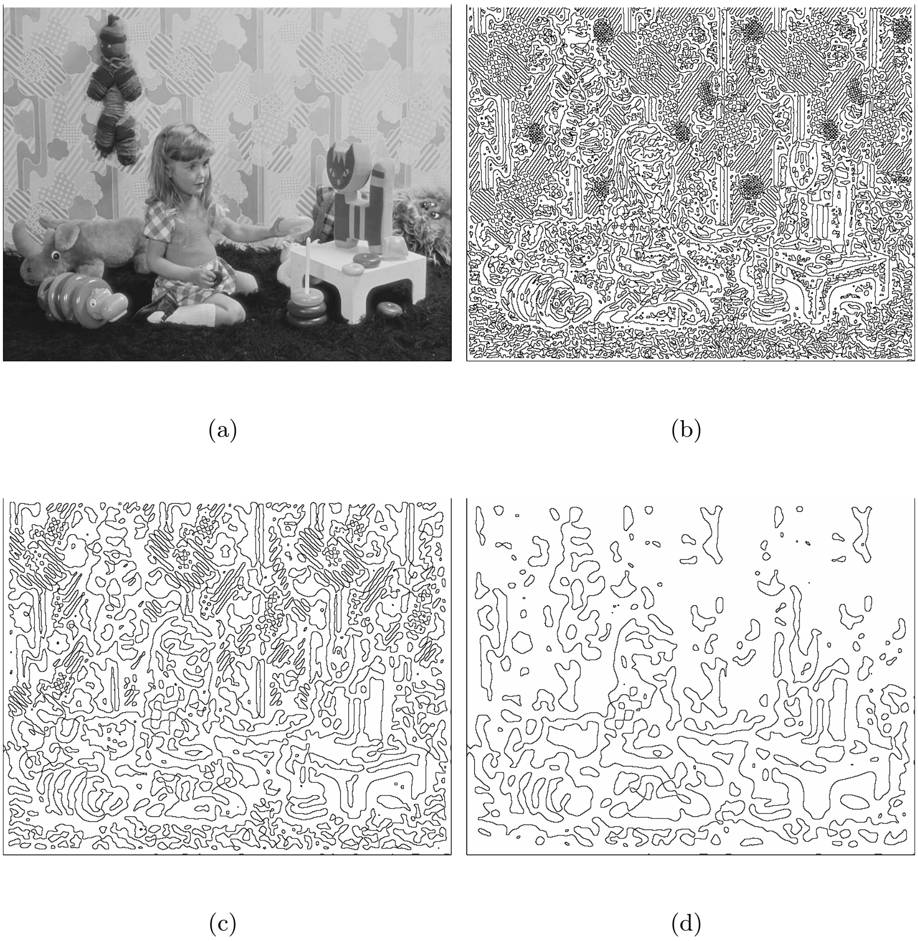


(a) (b)

Figure 2.5: Surface plots of the Laplacian of Gaussian kernels for σ equal to 1 (a) and 4 (b).

is placed on the shape of the lines, they tend to form closed contours and may therefore potentially define a segmentation. However, closer inspection reveals that for large σ the edges are very smooth causing discrepancies between 'edge' states and object boundaries at sharp corners, whilst a large number of insignificant edges are detected when σ is small, the texture of the carpet in the image of 'girl' for example.

In conclusion, whilst many techniques exist of the detection and refinement of edges in an image, those based upon the first derivative suffer badly from noise and quantisation error and fail to produce the closed contours necessary for defining a segmentation. This is due in part to the fact the operators which estimate the image gradient are relatively insensitive to soft edges which are more likely to occur than the model step edge. Second order gradient methods have a tendency to produce thin, closed contours, but there is a trade of between the accuracy of the edge locations and the noise rejection.



(a)

(b)

(c)

(d)

Figure 2.6: The original image of girl (a) and the zero-crossing detection of the Laplacian of Gaussian with σ equal to 2 (b), 4 (c) and 6 (d)

2.3 Region Based Segmentation Methods

The techniques discussed in the previous section aimed to define regions of the segmentation by locating their boundaries. This section introduces a number of established methods which define regions directly by grouping pixels according to the properties of the local neighbourhood. Region based methods have a number of advantages over both global and edge based techniques; they are capable of producing connected regions without the problems associated with broken or poorly located edges and are also inherently robust to the presence of noise. We begin by introducing the general philosophy of region based segmentation techniques and then describe the established methods of region splitting, split-and-merge and region merging giving examples. Also included is an introduction to connected and morphological operators which, though not traditionally considered region based segmentation techniques share many properties with region merging algorithms without the high computational cost.

The aim of all region based techniques is to produce regions which satisfy a homogeneity criterion, the general form of which is given by Equation 2.6, in which $H(R_i)$ is a measure of homogeneity evaluated over region R_i and T is a threshold above which a region is regarded as homogeneous.

$$H(R_i) \geq T \quad i = 1, 2, \ldots S \tag{2.6}$$

Typical measures of homogeneity include reciprocals of the variance σ given by Equation 2.7 or the range calculated by subtracting the smallest value in the region from the largest.

$$\sigma(f) = \frac{1}{|R_i|} \sum_{x,y \in R_i} (f(x,y) - \bar{R_i})^2 \tag{2.7}$$

Where $|R_i|$ is the number of pixels and $\bar{R_i}$ is the mean grey level in region R_i. The reciprocal of these measures is taken so as to arrive at inequalities of the form of Equations 2.8 and 2.6. Though the result of a regime which sought only to satisfy Equation 2.6 would, by definition, describe a valid partial segmentation, many region based segmentation techniques aim for the segmentation to be maximal in that the merging of two adjacent regions will necessarily result in a region for which the homogeneity criterion does not hold. Such methods aim to satisfy the expression

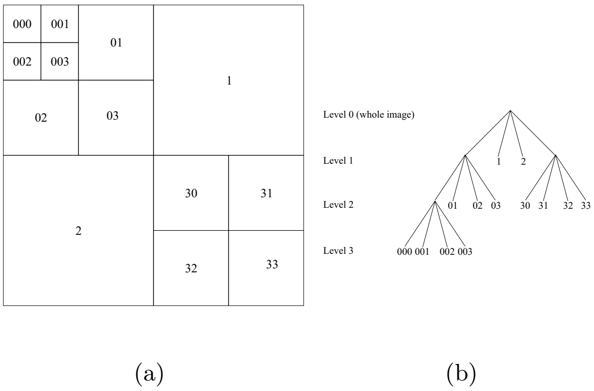


Figure 2.7: An example image segmentation with labelled regions (a) and the corresponding quadtree (b)

which is given in general from by Equation 2.8.

$$H(R_i \cup R_j) < T \quad i \neq j \quad \text{and} \quad R_i \quad \text{adjacent to} \quad R_j \tag{2.8}$$

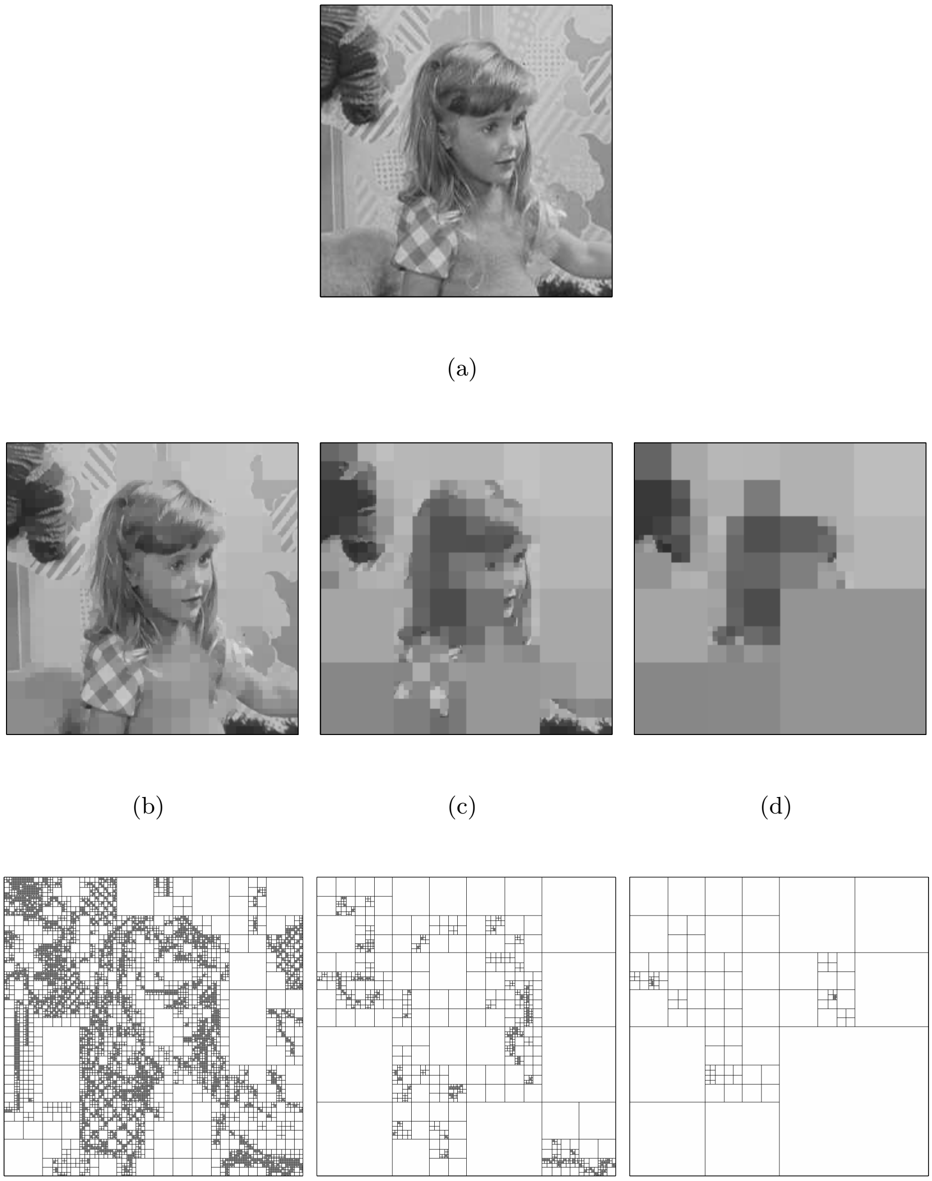
Region based segmentation techniques arrive at a solution by continuously evaluating $H(R_i)$. Splitting algorithms first evaluate the homogeneity of the entire image and if it falls below the threshold, the image is split into four quadrants. This process it then repeated for each of the regions until all regions satisfy Equation 2.6. So called split-and-merge algorithms add a merging stage in order to achieve a result which satisfies Equation 2.8. Conversely, region merging methods begin with an initial partition for which each region is already homogeneous, traditionally these partitions represent individual pixels. Partitions are then merged until Equation 2.8 is satisfied for all adjacent regions. We now discuss the methods of splitting, split-and-merge and region merging in detail.

Region splitting and split-and-merge algorithms can be conveniently described by the quadtree representation (Figure 2.7). At the top-most layer of the tree (layer zero) is a single region consisting of the entire image. A homogeneity threshold T is usually chosen such that the whole image is not regarded as homogeneous in that it generally does not satisfy Equation 2.6. As such, the region is split into four equal quadrants, each of which is described by a node in the next layer of the quadtree (layer one). In our example, the homogeneity of regions 1 and 2 (Figure 2.7(a)) satisfies Equation 2.6 and as such the associated nodes have no further children and are referred to as leaf nodes. Regions 0 and 3 of our example fail satisfy the homogeneity test and are again split into four quadrants, 00-03 and 30-33. The process of splitting into quadrants regions for which the homogeneity criterion is not met continues until all leaf nodes of the quadtree are associated with regions of the image which satisfy Equation 2.6 and are therefore regarded as homogeneous. The regions represented by the leaf nodes correspond to segments of the image as calculated by a pure splitting regime. The choice of threshold T, will clearly effect the number of regions in the resulting segmentation, a very high threshold will produce few, large regions, whilst a lower value will produce more, smaller regions.

Figure 2.8(a) shows a section of the image 'girl', for the full image see Figure 2.6(a). Figure 2.8(b) is the result of splitting the image using the quadtree representation described above with a homogeneity criterion which requires that the variance of the quadrant in terms of grey level must be below 100 [1]. Figure 2.8(e) shows the outline of the regions of the segmentation. Figures 2.8(c) and 2.8(d) show the results of using a variance threshold of 500 and 1000 respectively with Figures 2.8(f) and 2.8(g) showing the respective region outlines. For all three segmentation images, the grey level for each segment has been determined by the mean of the original image inside that segment. Such images shall hereafter be referred to a mean replaced images. For variance threshold of 100, the image appears slightly blocky with a loss of textural information and detail in some areas, for example the wallpaper and the girl's clothing, an effect which can be more easily observed in the outline image of Figure 2.8(e). The outline image clearly shows that the segmentation has a large number of small segments in areas of high contrast, (and therefore grey level variance) such as the girl's hair and eyes, whilst regions of the image with low contrast such as the wallpaper are represented by larger blocks.

Increasing the threshold to 500 results in an increase in the average block size, though there remain a large number of small segments at high contrast discontinuities such as the foot of the puppet (top left hand corner), the meeting of carpet and wallpaper in the lower right and the features of the girls face. At the same time, regions of the image which are relatively plain, such as the jumper of the girl and the back of the stuffed toy in the lower left corner, have been included in single segments of the result. Though this is desirable, it is unfortunate that the rectangular nature of the

[1] The units of grey level variance are the square of the units of the intensity function and will therefore depend on a number of factors relating to the image acquisition.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 2.8: A section of the original image ‘girl’ (a), the segmentations produced using the splitting algorithm with a variance threshold of 100 (b), 500 (c) and 1000 (d) and the region outlines thereof; (e), (f) and (g) respectively.

segments prevents the segment from masking the region perfectly.

Finally, the result of using a variance threshold of 1000 given in Figure 2.8(d) and 2.8(g) show a further increase in the average size of segments. It is clear though that the segments bear little if any correlation with objects in the image. Though this is not necessary for a partial segmentation, it is often desirable. It should be noted that while segments are homogeneous and thus satisfy the expression in Equation 2.6, there is no guarantee that the expression in Equation 2.8 is met by all adjacent regions and thus there is the potential for a single homogeneous object to be represented by more than one segment, particularly for strict homogeneity criteria such as $\sigma < 100$. In order to achieve a result which satisfies Equation 2.8 one can introduce a merging stage in which adjacent regions whose union meets the homogeneity criterion are merged.

Returning to the quadtree representation, addition of a merging stage allows regions represented by leaf nodes of different parents to merge provided the homogeneity criterion of the image function under the region of support of the resulting region is met. Figure 2.9 shows the results of such a scheme using a variance threshold for the homogeneity criterion as in the previous example. The same thresholds have also been used such that the segmentation images due to the split-and-merge algorithm can be regarded as merged versions of those of Figure 2.8. Close inspection of Figures 2.8(g) and 2.9(f) reveals that all the boundaries of the outline image for the split-and-merge algorithm are present in the pure splitting outline. Differences arise because of the boundaries removed in the merging stage. This holds for all variance thresholds.

The addition of a merging stage has unsurprisingly increased the average size of the segments for a given variance threshold. It has also improved to some extent the correlation of segment boundaries with the boundaries of objects in the scene by allowing segments to have more complex shapes. Unfortunately, the improvement is not sufficient, segment boundaries are still very blocky and segments have a tendency to include more than one distinct region of the image. Of course, it is not necessary to use the same homogeneity criterion for the merging stage as for splitting. Though the resulting method is not strictly regarded as split-and-merge as the result is no longer maximal with respect to a single homogeneity criterion, the results from

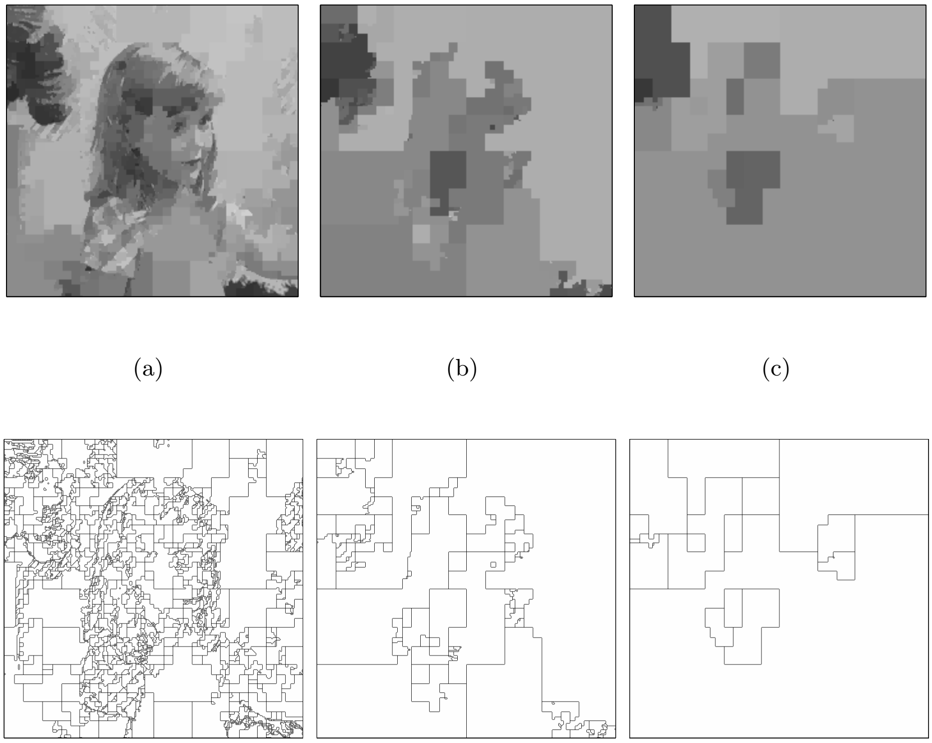
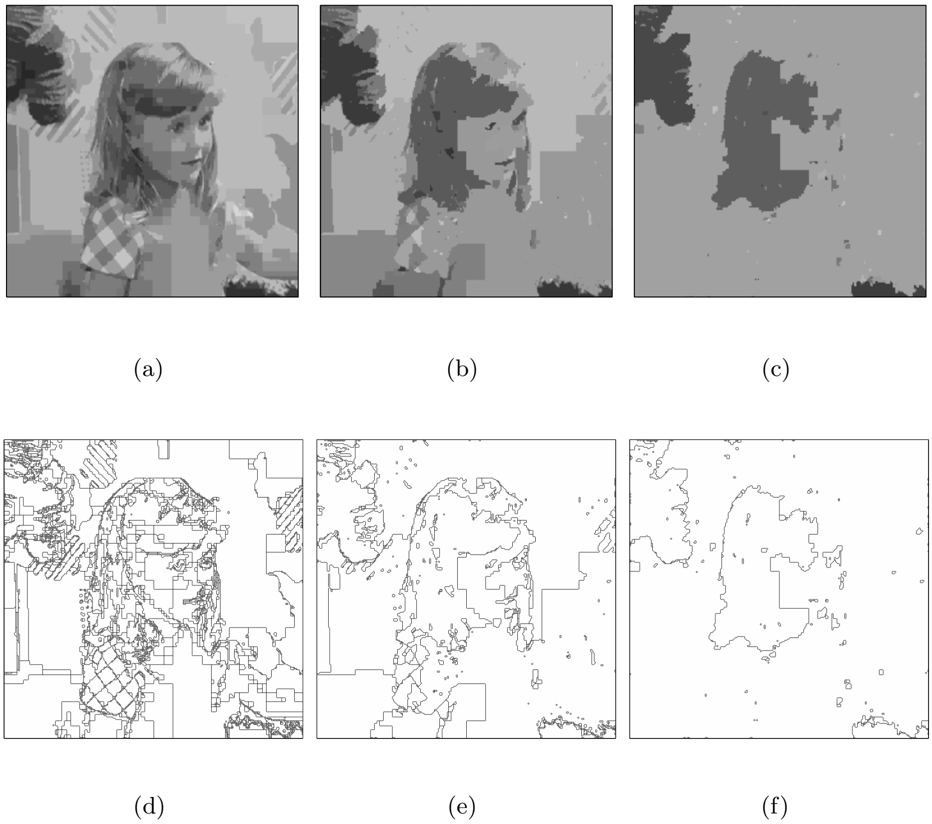


Figure 2.9: The resulting segmentations of the section of girl given in Figure 2.8(a) using a split-and-merge algorithm using a variance threshold of 100 (a), 500 (b) and 1000 (c).



(a) (b) (c)

(d) (e) (f)

Figure 2.10: The segmentations produced using the split-and-merge type algorithm with a variance threshold of 100 and mean difference threshold of 10 (a), 20 (b) and 30 (c) along with the region outlines thereof; (d), (e) and (f) respectively.

this scheme are often preferable to those which are based on a single homogeneity criterion.

Figure 2.10 shows the results of a split-and-merge type algorithm in which the standard splitting technique used in the previous examples is followed by a merging step in which regions are merged provided the mean grey levels of the image function inside the regions are sufficiently close. The results given use a variance threshold of 100 and mean difference thresholds of 10, 20 and 30 respectively. It is clear from these results that merging using the difference in mean values results in more realistic region boundaries. In order to increase the smoothness of the resulting contours using this method, the threshold for the variance in the splitting stage can be decreased. The result is a more stringent homogeneity criterion using which the splitting stage produces more, smaller regions. Indeed, in the limit as the homogeneity criterion requires that regions be completely flat with respect to grey level, the splitting phase will result in regions comprising single pixels and the split and merge technique described becomes a pure merging method, it is toward such methods that we now turn our attention.

Region merging techniques have direct comparisons with region growing methods [5] in which a small seed region of the image is grown by iteratively examining the neighbours of each pixel in the region and assessing them for inclusion in the region. Such region growing methods are used extensively for the interactive segmentation of single objects in a scene whereby the user selects a region inside the object to be segmented. Pixels which neighbour the selected region are assessed for inclusion based upon some measure of the likelihood that they belong inside the region. Two such schemes are referred to single linkage and centroid linkage region growing techniques [2]. Single linkage region growing will include in the region a neighbouring pixel if the difference in grey level between that pixel and the nearest neighbour inside the region is less than a given threshold. Centroid linkage region growing includes in the region any neighbouring pixel whose grey level is close enough to the mean of all pixels already included in the region. The growing of the region ceases once no more neighbours can be included. Clearly, the growing of a single region is not sufficient for automated image segmentation and as such an arbitrary number of regions must be grown in such a way that they do not overlap, but fill the entire domain of the image. The strategy used in [2] involves scanning the image, left to right, top to bottom. In the case of a single linkage regime, if the difference between the current pixel and any of its causal neighbours (those which have already been scanned) is less than a given threshold then the pixel is included in the region of the corresponding pixel. If the threshold is met by more than one pixel, then all regions to which those pixels belong are merged. If no neighbouring pixels are similar enough, then the pixel defines a new region. The centroid linkage regime is similar but rather than comparing the grey levels of individual pixels, the grey level of the current pixel is compared with the mean grey levels of the regions associated with the causal neighbours. Figures 2.11 and 2.12 show examples of the application of single and centroid linkage region growing techniques for the section of the image 'girl' presented in Figure 2.8(a). It is clear that region merging has improved the shape of the contours and even for these relatively simple examples,

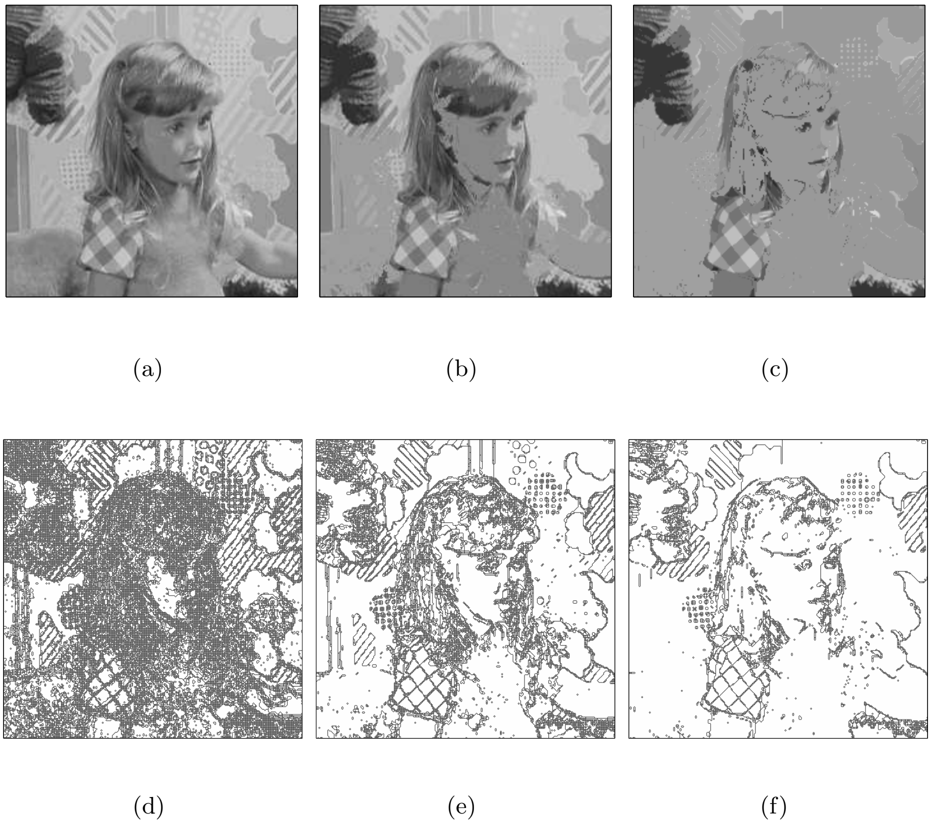
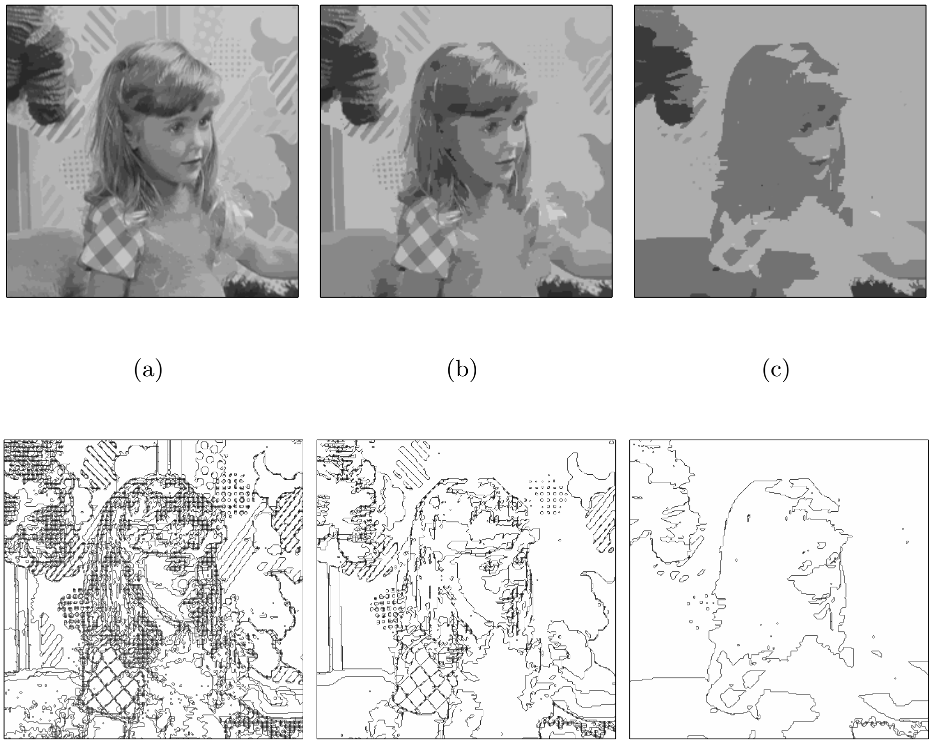


Figure 2.11: Single linkage segmentation of a section of the image 'girl' using a grey level threshold of 2 (a), 4 (b), and 6 (d) along with the corresponding outline images (d), (e) and (f) respectively.



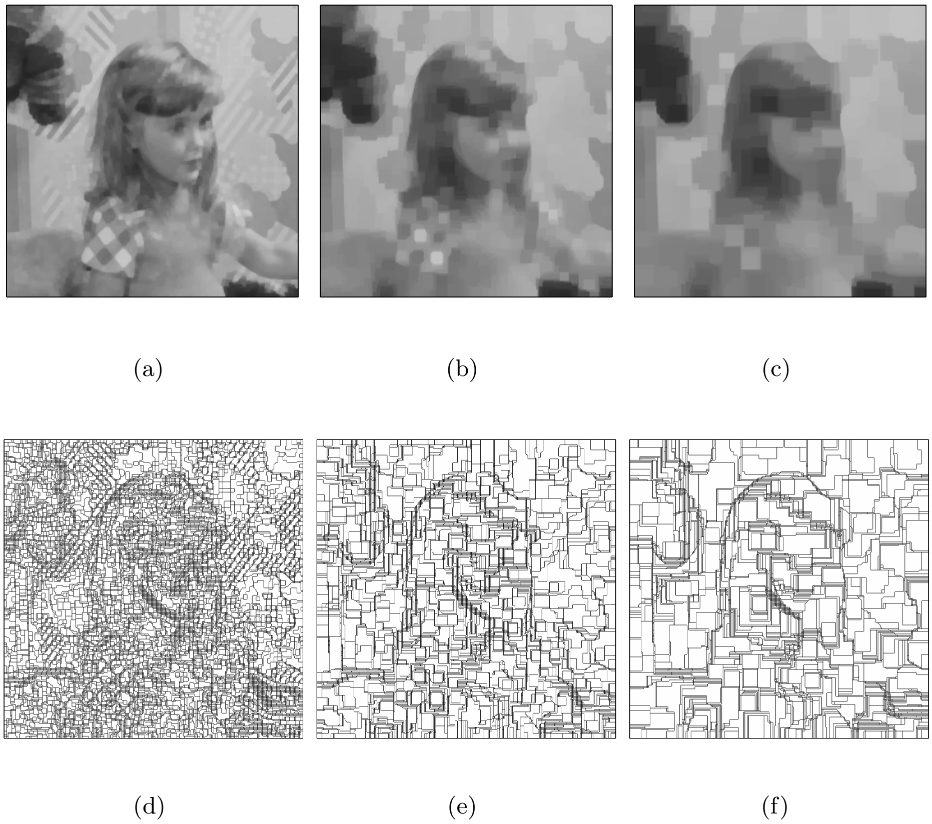
(a) (b) (c)

(d) (e) (f)

Figure 2.12: Centroid linkage segmentation of a section of the image ‘girl’ using a grey level threshold of 5 (a), 10 (b), and 20 (d) along with the corresponding outline images (d), (e) and (f) respectively.

the regions correspond relatively well with homogeneous regions of the image. More complex statistical measures of a regions properties can be used to assess the likelihood of a pixel's inclusion and whilst these may produce more desirable results in which segments have improved correlation with object regions, the computational load increases significantly. This is a major drawback of pure region merging techniques which start at the pixel level. One method of improving the efficiency of region merging algorithms is to begin not at the pixel level but with some arbitrary partition as in the splitting regime described previously. One such partition is that due to the watershed algorithm discussed in the following section, another is to partition the image into *flat zones* or regions in which the grey level is constant. It is the partition of flat zones which forms the basis of connected operators a set of operators capable of image simplification and for which highly efficient algorithms have been developed [6–9]. A particular set of connected operators referred to as morphological operators appear earliest in the literature and are quoted as being capable of removing image detail whilst preserving essential shape information Haralick1987. Morphological operators for image processing have received considerable attention in the literature [10–14] and more recently their relationship with region merging algorithms has been studied [15]. The dual operators of *opening* and *closing* suppress local maxima and minima respectively using order statistic filters [12]. A full description of morphological and connected operators is given in the following Chapter but the results for the open-closing (an opening followed by a closing) of the image 'girl' for square structuring elements of three different sizes are presented in Figure 2.13. The result of the open-closing is that both maxima and minima are removed from the image leaving behind a smoothed version of the original. This smoothing is very different from the smoothing achieved by linear filtering in that much of the edge information is preserved. Segments may then be taken to be regions in which the grey level is constant.

Region merging methods do not use the quadtree structure and as such do not suffer from the problems of splitting or split-and-merge techniques. Because they are region based they do not suffer the problems with regard to the sensitivity to noise of edge based techniques. Of all the partial segmentation techniques discussed in this chapter, region merging methods appear to have all the desirable properties of a segmentation technique capable of fulfilling the required criteria. A full review of these operators including further examples of their use in image segmentation is



(a) (b) (c)

(d) (e) (f)

Figure 2.13: The results of open-closing a section of the image ‘girl’ using a square structuring element of size 3 × 3, (a), 5 × 5, (b) and 7 × 7 (c) and corresponding outlines (d), (e) and (f) respectively.

reserved for subsequent chapters.

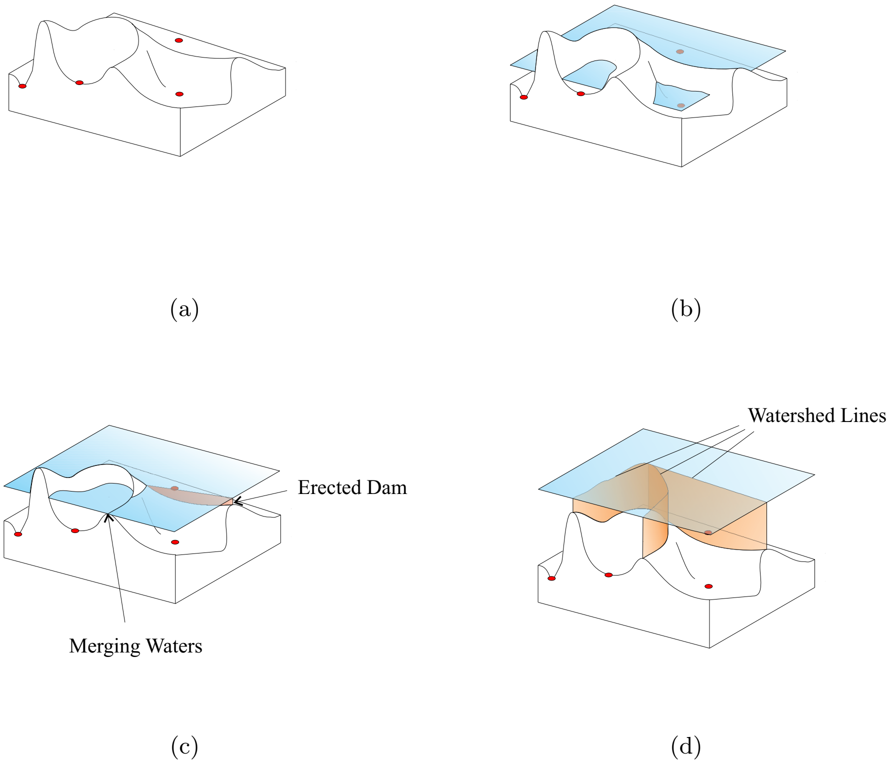
The final section of this chapter introduces the watershed transform which can be regarded as both a method of edge detection [16] and region based segmentation [17].

2.4 The Watershed Transform

The watershed algorithm is the morphological operator most commonly associated with image segmentation [17]. In geographical topography, a watershed is a ridge which divides regions drained by different river systems. The North American Continental Divide provides a classic example of a watershed line dividing the catchment basins of the Atlantic Ocean to the west and the Pacific Ocean to the east. In mathematical morphology, the watershed transform identifies the catchment basins of an image considered as a topographical relief with height proportional to grey level. The definition of a watershed for both continuous and digital functions based upon topographical distance is given in [18], which states that each catchment basin is associated with a regional minimum[2] of the image grey level and consists of the set of points which are topographically closer to the associated minimum than any other.

The authors of [17] present a more pragmatic approach referred to as the flooding scheme in which the image represented as a topological relief is pierced at each of its regional minima (Figure 2.14(a)) and submerged into a 'lake' thus filling each catchment basin from its associated minima (Figure 2.14(b)). The speed at which submersion takes place is slow enough for the water to have reached the same level in each basin, though methods in which the fluid is considered viscous have been proposed [19]. To avoid catchment basins merging, a dam is built everywhere waters associated with different minima meet (Figure 2.14(c)), these dams correspond with the crests of the terrain and after complete submersion of the relief the lines described by the dams in aerial view represent the watersheds (Figure 2.14(d)). An algorithm for the fast computation of watershed lines using the flooding scheme is presented in [20] and it is this algorithm which is used throughout this research.

[2] A regional minimum is a region of uniform height without lower neighbours.



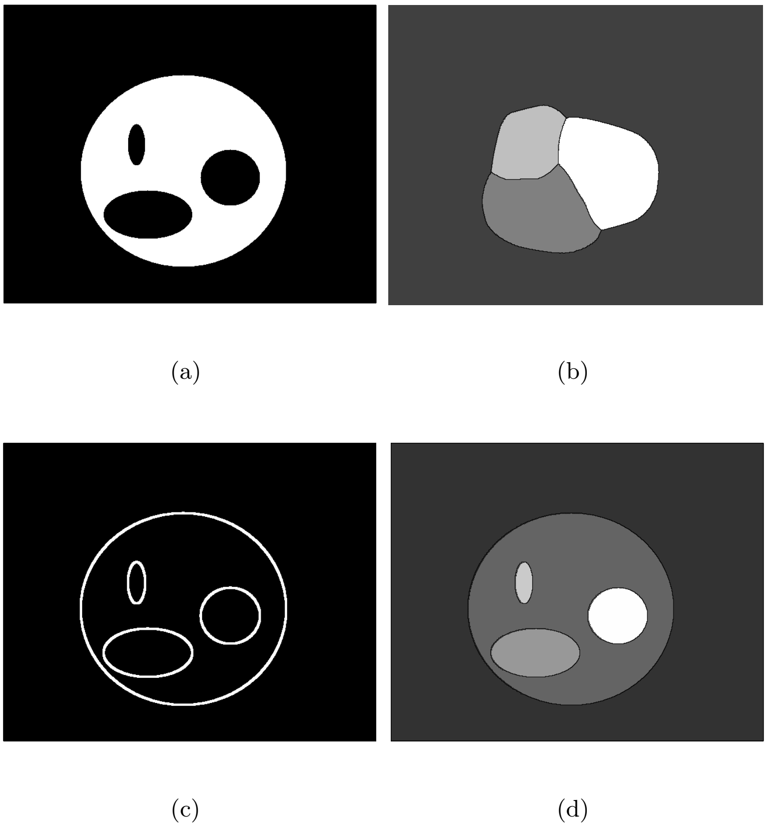
(a)

(b)

(c)

(d)

Figure 2.14: An arbitrary image function viewed as a relief with regional minima marked with dots (a), after partial flooding (b) and after continued flooding (c).



(a) (b)

(c) (d)

Figure 2.15: A synthetic black and white image (a), and the watershed thereof (b). The morphological gradient of the image (c), and corresponding watershed (d).

Figure 2.15(a) shows a simple synthetic image with white taking a high value and black a low value. As such, the three black regions at the centre of the image and the background represent regional minima and therefore each have a corresponding catchment basin. Figure 2.15(b) shows the watershed of the image, black representing the watershed lines and each grey level uniquely defining a catchment basin. It is clear from Figure 2.15(b) that the watersheds lie entirely within the white region and describe lines which are in a sense optimal in that they remain equidistant from all influencing minima.

The example given above shows how an image can be segmented into regions which correspond to topological catchment basins. For the partial segmentation of images however, the watershed algorithm is more frequently used on the gradient estimate of the image [17]. Considered topologically, the gradient of an image takes high values where the grey level changes rapidly and low values in flat regions. Since it is likely that the grey level gradient will be high at object boundaries, locating the crests of the gradient image should locate object edges in the image. Figure 2.15(c) shows the morphological gradient of the image given in Figure 2.15(a). For a given image function, $f(x, y)$, the morphological gradient $\mathrm{grad}_{\mathrm{M}}(f)$ is calculated by subtracting the grey scale erosion from grey scale dilation of the image (see Chapter 3,) and hence, each pixel of the result is assigned a value equal to the difference between the maximum and minimum values in the neighbourhood described by the structuring element k.

$$\mathrm{grad}_{\mathrm{M}}(f, k) = f \oplus_{\mathrm{g}} k - f \ominus_{\mathrm{g}} k \tag{2.9}$$

Where $\oplus_{\mathrm{g}}$ and $\ominus_{\mathrm{g}}$ are grey scale dilation and erosion respectively and are defined in the following chapter. In practice, this method of evaluating the gradient of an image differs little from those presented in the previous sections but is used here to follow [17]. In our example, the structuring element used is a square of size 5×5 pixels. Note that the image given in figure 2.15(c) has been normalised in order to make visible the regions of high gradient which in this idealised case correspond perfectly with the edges of the black ellipses. Performing the watershed operation on the morphological gradient yields the image given in Figure 2.15(d) from which it is clear that the regions defined by the catchment basins correspond perfectly with the regions of the image.

It is rarely, if ever, the case that real images which we may wish to segment are as free from noise and quantisation error as was the previous example. As demonstrated previously in the case of edge detection, gradient operators are highly sensitive to such disturbances in that a large number of peaks and troughs will appear in the gradient image frequently resulting in vast over-segmentation by the watershed transform. Figure 2.16(a) shows the image of Figure 2.15(a) with the addition of Gaussian distributed white noise with a mean of 0 and a variance of 0.01 times the range of the image which in this case is 0-255. Figure 2.16(b) shows the watershed of the morphological gradient of the noisy image, note that the shading of individual regions has been omitted due to their number; the watershed lines are shown in black. It is clear from this figure that the addition of noise has resulted in far

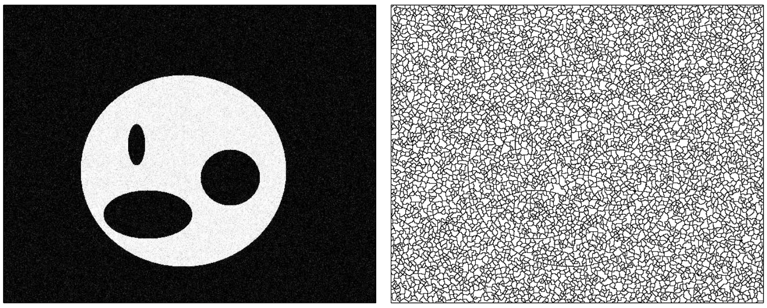


Figure 2.16: The image of Figure 2.15(a) corrupted by Gaussian noise (a) and the outline of the watershed of the morphological gradient thereof (b).

too many regions being identified. Close inspection shows that along with all the boundaries of unwanted regions, the edges of the objects have been successfully identified.

Methods for reducing the over-segmentation of the watershed transform can be summarised under the following titles:

- Watershed From Markers [17, 21]
- Region Merging [22]
- Improved Gradient Estimation [3, 23, 24]

The problem of over-segmentation due to the sensitivity of gradient operators is tackled by the authors of [17] in their study of electrophoresis gels. It is observed that the gradient of the noise contains a large number of regional minima, each of which necessarily defines a catchment basin and hence a region of the segmentation achieved by means of the watershed transform. The solution to over-segmentation in terms of the immersion model is to pierce not all regional minima, but only those of greatest significance (being the minima associated with the objects and the background). The flooding simulation continues as described previously, but water breaching a divide between two minima is allowed to fill the new minima provided it does not already contain water originating from a different pierced minima. Dams and hence watersheds are constructed at the meeting point of waters from different holes and hence the number of regions defined in the result is equal to the number of original holes. This type of watershed algorithm is referred to as a watershed from markers and is described in detail in [17]. The drawback of this method of reducing the number of regions identified in the segmentation is that more processing or possibly user interaction is required to identify the markers and this processing will be highly dependant upon the application.

The authors of [22] propose a scheme in which the image is subjected to noise reduction prior to calculation of the gradient. It is also proposed that the gradient image be thresholded so as to further reduce the tendency to over-segment. Problems incurred in the thresholding of the gradient image have been addressed in previous section and similar effects are observed in thresholding the gradient image prior to calculation of the watersheds. This problem is addressed in [24] in which the response of the morphological gradient to soft edges is enhanced whilst leaving the response to noise relatively low by taking a multi-scale approach. The authors go on to propose methods by which the resulting regions of the watershed transform can be merged with their neighbours based on the evaluation of a homogeneity criterion as in standard region merging algorithms. The improvement of the gradient image and the application of connected operators to region merging of the watershed transform form the basis of Chapter 5 of this thesis.

2.5 Summary

In this chapter, we have introduced the concept of an arbitrary partial segmentation being a complete set of disjoint, non-overlapping regions (Definition 1) inside each of which the image is regarded as homogeneous with respect to some measure based up the grey level information. We have seen how split-and-merge techniques define a partition of the image domain in the splitting stage. A model of the grey level (such as the mean) is then associated with each region and regions are then merged based upon the difference in these models. Region growing methods begin with the initial partition defined by the grid of the domain, each pixel being represented by a single partition. The merging order, being the order in which partitions are analysed for merging, has in our examples been that of the raster scan and merging is again based upon some measure of distance in the grey level domain. In the case of centroid linkage region growing a model for each region, again based on the mean, is used in the merging criterion. Connected operators begin with the partition of flat zones. In the opening (resp. closing) of an image, regional maxima (minima) are merged with their lower (higher) neighbours if they are removed in the erosion (dilation) of the image by the structuring element used. Finally, the watershed algorithm has initial partitions which correspond with pixels, regional minima (or markers in the case of watershed from markers) are then grown based upon a model of the partition being the notion of a unique catchment basin and a merging criterion based upon the image gradient. In the sequel a unifying framework, the lattice of partitions, which incorporates all the region based segmentation methods described thus far. Particular attention is given to morphological and connected operators as these, it will be shown, can be used to obtain segmentations possessing the properties we require.

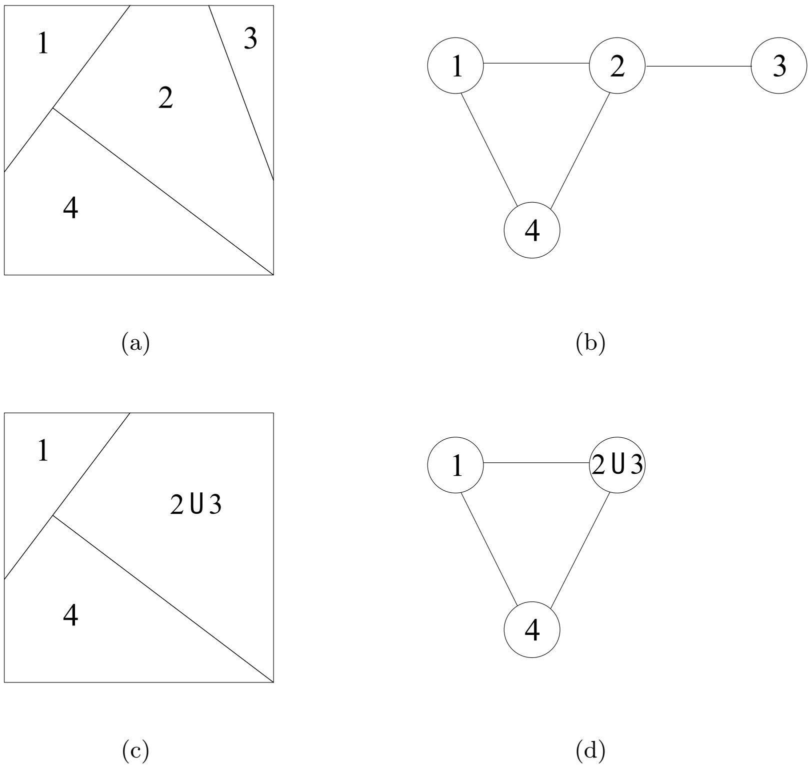
Chapter 3

The Lattice of Partitions, Morphological and Connected Operators

In the previous chapter we introduced the concepts of segmentation, region based spatial clustering, morphological and connected operators and the watershed algorithm. All of the techniques utilise the same basic merging strategy consisting of the following components first proposed in this context in [25]:

- An initial partition
- A merging order
- A region model

This basic framework allows all the methods to be conveniently modelled using the *region adjacency graph* (RAG) with regions of the partition represented by nodes and their connectivity to other regions by links between these nodes. Using the RAG as a basic model, this chapter unifies these region based methods under the framework of the *lattice of partitions* and develops the methods of mathematical morphology and connected operators for image segmentation.



(a) (b) (c) (d)

Figure 3.1: A simple partition (a) and the associated region adjacency graph (RAG) (b) along with the partition (d) and RAG (d) after merging regions 2 and 3.

Figure 3.1 shows a simple partition and the associated RAG before and after the merging of two regions. Initially the nodes of the RAG may represent regions defined after the splitting stage of a split-and-merge algorithm, the pixels in the case of merging and watershed techniques or flat zones for connected operators. As regions are merged nodes of the tree are also merged and the corresponding links removed. The RAG model for region merging allows for the conception of a *partition lattice*. A full theoretical treatment of complete lattices can be found in [26]. The definition of a lattice is based upon the notion of a *partially ordered set*.

Definition 2 (The partially ordered set) Given a non empty set $\mathcal{L}$, a binary relation $(\leq)$ on $\mathcal{L}$ is called partial ordering if the properties of reflexivity, anti-symmetry and transitivity hold for any pair of elements. The set $\mathcal{L}$ for which the partial ordering holds, denoted $(\mathcal{L}, \leq)$ is called a partially ordered set.

Partial ordering (as opposed to complete ordering) implies that while the binary order operation may hold for some element pairs, any two elements may be such that neither is either greater than, less than or indeed equal to the other. This rather paradoxical statement can be qualified by way of an example.

Consider the set of all integer numbers $\mathbb{Z}$. We define the ordering $x \leq y$, where x and y are elements of $\mathbb{Z}$, in the usual way. It is clear that for any pair of elements either $x \leq y$ or $y \leq x$ and as such the ordering is complete. If however we extend the set to $\mathbb{Z}^N$ and define the ordering relation such that $(x_1, x_2, \dots x_N) \leq (y_1, y_2, \dots y_N) \iff x_i \leq y_i \;\forall\; i$ then there exist pairs of elements for which the ordering does not hold. For example, in the case $\mathbb{Z}^2$ we can define two integer couples $\mathbf{x} = [1, 2]$ and $\mathbf{y} = [3, 1]$. Since for the first element, $x_1 < y_1$ and for the second $x_2 > y_2$ we can say that neither couple is greater than the other and they are clearly not equal.

A complete lattice is a partially ordered set, each subset of which possesses a supremum $\vee$ and infimum $\wedge$, being a lowest upper and greatest lower bound respectively.

Definition 3 (The lattice) A partially ordered set $\mathcal{L}$ is called a lattice if every finite subset of $\mathcal{L}$ has a supremum and infimum. Furthermore, the lattice is a *complete* lattice if every subset of $\mathcal{L}$ has a supremum and infimum.

We now define our partition lattice as being the set of all possible partitions $\mathcal{P}_i$ of the image domain, the set of regions of partition $\mathcal{P}_i$ being denoted as $\{\mathcal{R}_i^n\}$. The ordering of the lattice is given according to the following relation:

Definition 4 (Order relation for partitions) Partition $\mathcal{P}_i$ is smaller (or *finer*) than partition $\mathcal{P}_j$ if all points belonging to a given region of partition $\mathcal{P}_i$ also belong to a single region of region of partition $\mathcal{P}_j$. Formally, $\mathcal{P}_j \leq \mathcal{P}_i \iff \forall\; x, y \in \mathcal{R}_i^n,\; \exists\; m$ such that $x, y \in \mathcal{R}_j^m$.[25]

It should be noted that since the ordering relation is partial, two arbitrarily chosen partitions may not be comparable (as in the example above). Partitions which can be compared however represent a hierarchy of partitions, the largest consisting of a single region (the entire image) and the smallest being the partition whose regions each correspond to a pixel in the image. Having stated that the lattice of partitions is indeed a lattice the interpretation of a infimum and supremum of a subset thereof must be clarified. Any two pixels x, y belong to the same region of the supremum of a set of partitions if they also belong to the same region in at least one other other partitions in the set.

Definition 5 (The supremum of a set of partitions) A partition is a supremum of a set of partitions $\vee \{\mathcal{P}_i\}$ if and only if all pairs of points in a given region of the supremum also appear in the same region of at least one of the partitions in the set.

Conversely, the infimum of a set of partitions is such that it falls lower than or equal to the lowest partition of the set according to the order relation, i.e. $\wedge \{\mathcal{P}_i\} \leq \mathcal{P}_i \; \forall \; i$.

Definition 6 (The infimum of a set of partitions) A partition is an infimum of a set of partitions $\wedge \{\mathcal{P}_i\}$ if and only if all pairs of points in a given region of the infimum also appear in the same region of all the partitions in the set.

Note that the resulting partition need not be in the original set. By duality the supremum of a set of partitions can be defined thus [26]: In a similar but dual statement to that made previously we can say that whilst the supremum need not be a partition of the original subset, it is at least equal to and potentially greater than the highest partition therein in terms of the order relation.

Having defined the lattice of partitions, it is now possible to discuss the properties of operators acting on the elements thereof. An operator ψ acting on a partition $\mathcal{P}_i$ may have following properties:

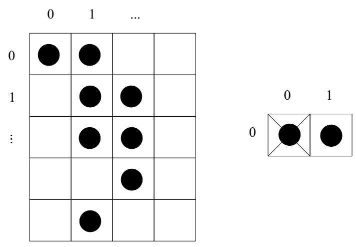
- ψ is *increasing* if and only if it is order preserving in that order of the outputs is the same as the order of the inputs. More formally, $\forall \mathcal{P}_i, \mathcal{P}_j,\ \mathcal{P}_i \leq \mathcal{P}_j, \Rightarrow \psi(\mathcal{P}_i) \leq \psi\left(\mathcal{P}_j\right)$. In the case of connected operators and region merging algorithms, this property is not easy to maintain.
- ψ is *anti-extensive* if the output partition is smaller (in terms of the order relation) than the input partition. i.e. $\mathcal{P}_i \leq \psi\left\{\mathcal{P}_i\right\}$. Thus an anti-extensive operator must split the partition such that more regions are present in the result than in the original. Conversely an *extensive* operator is one for which

the output partition is greater than or equal to the input, i.e. $\psi\{\mathcal{P}_i\} \leq \mathcal{P}_i$ requiring that the operator merges regions of the partition upon which it acts.

- ψ is idempotent if the result is such that a second application of the operator yields no further change in the partition. Formally: $\forall \mathcal{P}_i,\ \psi(\psi(\mathcal{P}_i)) = \psi(\mathcal{P}_i)$. The implication of this in terms of merging algorithms is a stable termination of the merging process.
- ψ is morphological if it is increasing and idempotent.
- ψ is an opening (resp. closing) if it is an anti-extensive (resp. extensive) morphological operator.

Thus we can describe the splitting stage of the split and merge algorithm as an increasing, anti-extensive, idempotent operator acting upon subset of the lattice of partitions. Similarly, the merging stage is an increasing, extensive, idempotent operator acting upon the same hierarchical subset of the lattice. Connected operators which we have described in terms of the partition of flat zones are both increasing and extensive, since they merge regions of the partition.

Having introduced a convenient framework for the discussion of morphological and connected operators, the remainder of this chapter is concerned with the introduction of these operators in the image domain. We begin by introducing the basic axioms of mathematical morphology in image processing, these being the binary set operations of erosion and dilation. We then extend these to grey scale images and show how they can be used sequentially to form openings, closing and the dual sequential open-closing and close-opening filters. These are then shown to have potential in image segmentation but for the constraint of the size and shape of the structuring element. Methods of overcoming this constraint, *geodesic transforms*, *attribute morphology* and *filters by reconstruction* are then introduced and defined under the title of connected operators.



(a) (b)

Figure 3.2: A simple binary image with fourground pixels denoted by black dots (a), and a binary structuring element with centre marked by a cross (b).

3.1 Mathematical Morphology for Image Processing

Consider the binary image X represented as a set of co-ordinate couples whose vectors positions in the digital Euclidean plane $(\mathbb{Z}^2)^1$ correspond with the pixels of the image regarded as foreground. This image representation allows for the manipulation of the image via common set operations such as union, intersection and set difference (see Appendix A). We define in the same way a *structuring element B* usually having a considerably smaller domain than the image under study. Figure 3.2 shows a simple binary image with foreground pixels marked with black dots, along with a 2 × 1 full structuring element with the cross marking the origin.

Definition 7 (Binary set dilation) The dilation of X by B, denoted $X \oplus B$ is the set of points given by the addition of all elements of X with all elements of B:

$$X \oplus B = \{c \in \mathbb{Z}^2 \quad | \quad c = x + b \text{ for some } x \in X \text{ and some } b \in B\} \tag{3.1}$$

This can also be described as the set of points described by the union of all translates of X by the elements of B, an operation frequently referred to as Minkowski addition [10, 13]. Formally:

$$X \oplus B = \bigcup_{b \in B} X_b \tag{3.2}$$

[1] Much of the theory presented here is applicable to continuous functions in many dimensions, however, we choose to restrict our attention to those in digital spaces and here in only two dimensions.

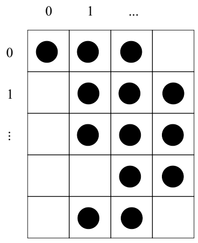


Figure 3.3: The binary image of figure 3.2(a) after dilation by the structuring element of figure 3.2(b).

Erosion is the morphological dual of dilation with respect to complementation in that the erosion of the foreground set X is identical to the dilation of the set describing the background or complement, X^c.

Definition 8 (Erosion of a binary set) Erosion of a binary image set X by the structuring element B is denoted $X \ominus B$ and is given by the following:

$$X \ominus B = \left\{ c \in \mathbb{Z}^2 \quad | \quad c + b \in X \quad \forall b \in B \right\} \tag{3.3}$$

This can be interpreted as the set of points in X for which the translation of the structuring element B_x is completely enclosed in X. Formally, this can be written using the intersection of translates:

$$X \ominus B = \bigcap_{b \in B} X_{-b} \tag{3.4}$$

This is not to be confused with Minkowski subtraction used by the authors of [13] and [14] in which the translation of the image set is by b and not $-b$. Figure 3.4 gives the erosion of the binary image set of Figure 3.2(a) by the structuring element of Figure 3.2(b).

The properties of erosion and dilation have been discussed extensively in the literature [10, 13, 14, 26]. It suffices here to say that dilation grows foreground regions whilst shrinking the background and conversely, erosions shrinks the foreground thus growing the background. They are not however inverse operators in that the dilation of an eroded binary set is not necessarily equal to the original set. Figure 3.5 shows the dilation of the binary image set given in Figure 3.4, itself the erosion of the original image set (Figure 3.2(a)). This alternating, dual operation is in fact a morphological opening.

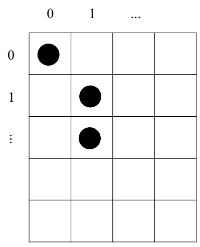


Figure 3.4: The binary image of figure 3.2(a) after erosion by the structuring element of figure 3.2(b).

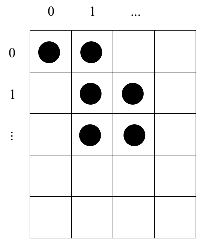


Figure 3.5: The dilation of the binary image set given by Figure 3.4.

Definition 9 (Opening) The opening of a set X by a structuring element B is denoted $X \circ B$ and given by:

$$X \circ B = (X \ominus B) \oplus B \tag{3.5}$$

The effect of opening a binary image is to remove foreground components which are completely removed by erosion and therefore cannot be reconstructed in the dilation. The dual operator (again with respect to complementation) is that of closing.

Definition 10 (Closing) The closing of a set X by a structuring element B is denoted $X \bullet B$ and given by:

$$X \bullet B = (X \oplus B) \ominus B \tag{3.6}$$

Being the dual of opening, the effect of closing is to add to the foreground set any components which are added in dilation and cannot be removed by subsequent erosion. Figure 3.6 shows a simple binary image along with the opening and closing thereof with a square structuring element of size 5×5. Note the removal of the thin foreground components in the opening and the thin background components in the

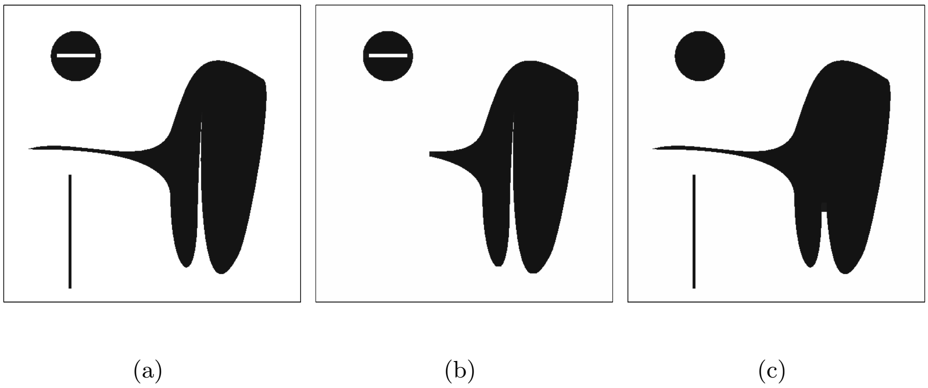


Figure 3.6: A simple binary image (foreground shown in black) (a), and the opening (b) and closing (c) thereof with a 5×5 square structuring element.

closing. We can in fact correlate the size of the removed components and the size of the structuring element used in the operation, though this relationship may not be immediately obvious. The erosion of a foreground component being at all points smaller than the structuring element in at least one axis results in its complete removal, regardless of its absolute size in pixels. Thus a vertically orientated line of width 2 pixels and length 100, will be completely removed by structuring element of dimensions 3×1. Hereafter, a feature of the image being 'smaller' than the structuring element will imply that it is at all points smaller than the structuring element in at least one axis.

In the extension of the operators of dilation, erosion, opening and closing to operate on grey scale images, it is reasonable to expect them to affect not foreground and background (since these are rather more complex notions) but on maxima and minima in the grey level function. The extension of the basic morphological operators to grey scale is a natural consequence of the umbra homomorphism theorem, a method described in [10] and [13]. It can be shown that the grey scale operators of dilation and erosion can be expressed as a maximum of sums and a minimum of differences respectively.

Definition 11 (Grey scale dilation as a maximum of sums) Given the grey scale image set f and the grey scale structuring element k, the grey scale dilation of f by k is given by:

$$(f \oplus_{\mathrm{g}} k)(\mathbf{x}) = \max_{\substack{\mathbf{u} \in k \\ \mathbf{x}-\mathbf{u} \in f}} \{f(\mathbf{x}-\mathbf{u}) + k(\mathbf{u})\} \tag{3.7}$$

Note the use of the subscript $_{\mathrm{g}}$ to denote that the operation is grey scale. A grey scale image set consists of elements which themselves have three associated values. The first two, denoted by the vector $\mathbf{x}$, represent the position in the image domain and the third, y, the grey level (see Appendix A). In the above definition, the notation $f(\mathbf{x})$ returns the grey level of the function f at location $\mathbf{x}$. We can define grey scale erosion:

Definition 12 (Grey scale erosion as a minimum of differences) Given the grey scale image set f and the grey scale structuring element k, the grey scale erosion of f by k is given by:

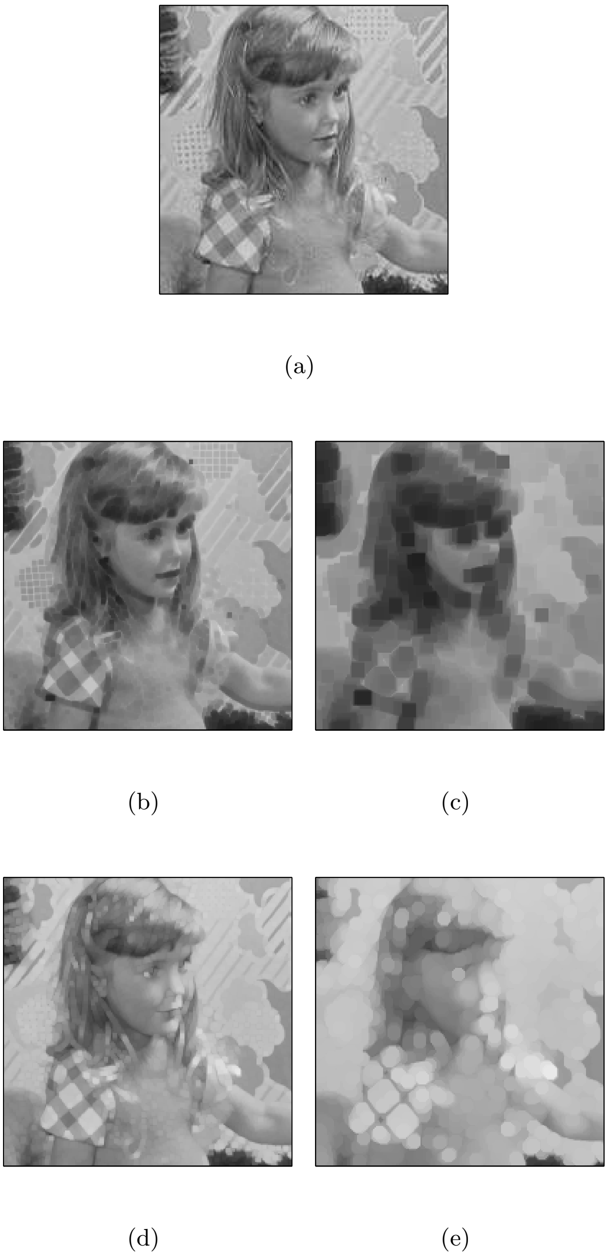
$$(f \ominus_{\mathrm{g}} k)(\mathbf{x}) = \min_{\mathbf{u} \in k} \{f(\mathbf{x}+\mathbf{u}) - k(\mathbf{u})\} \tag{3.8}$$

Note that for both erosion and dilation, the structuring element not only describes a neighbourhood as in the binary case, but also has a height. The use of non-flat structuring elements has been shown to be useful in some applications including the computation of the distance transform [8]. Such operators tend to introduce topological features into the image function, an effect which is undesirable for our purposes and as such we restrict our attention to flat structuring elements. The expressions for grey scale dilation and erosion thus reduce to the following:

$$f \oplus_{\mathrm{g}} k = \max_{\substack{\mathbf{u} \in K \\ \mathbf{x}-\mathbf{u} \in F}} \{f(\mathbf{x}-\mathbf{u})\} \tag{3.9}$$

$$f \ominus_{\mathrm{g}} k = \min_{\mathbf{u} \in K} \{f(\mathbf{x}+\mathbf{u})\} \tag{3.10}$$

These have been compared with order statistic, median and stack filters in [11]. The duality relationship for binary erosion and dilation holds for the grey scale case with the complementation of a grey scale image being inversion of the image function. Inversion of a grey scale image is computed by subtracting the image function from the maximum attainable value (255 in the unsigned 8-bit case) thus swapping bright for dark. Figure 3.7 shows a section of the image 'girl' eroded and dilated by square and circular structuring elements. It is clear from Figures



(a)

(b)

(c)

(d)

(e)

Figure 3.7: A section of the original image of ‘girl’ (a) along with the grey scale erosions with a square structuring element of sizes 3 × 3 (b) and 9 × 9 (c) and the grey scale dilations by a circular structuring element of radii 1 (d) and 4 (e).

3.7(b) and 3.7(c) that erosion increases the sizes of dark regions (regional minima of the image function) whilst decreasing the size of bright regions (regional maxima) by an amount proportional to structuring element size. The dilation of the image (Figures 3.7(d) and 3.7(e)) has the dual effect in that it decreases the size of regional minima whilst increasing the size of regional maxima. Different shaped structuring elements of equivalent size have been used in these examples to illustrate the effect of structuring element shape on the dilation. Note that in the dilation, regional minima have been expanded to take on the shape of a square whilst in the erosion, maxima have taken on a circular form. This is an unfortunate consequence of the standard morphological operators and will be discussed in more detail in the sequel.

Continuing the analogy with binary morphological operations, we can define grey scale opening and closing.

Definition 13 (Grey scale opening) The grey scale opening of an image function f by a structuring element k is given by:

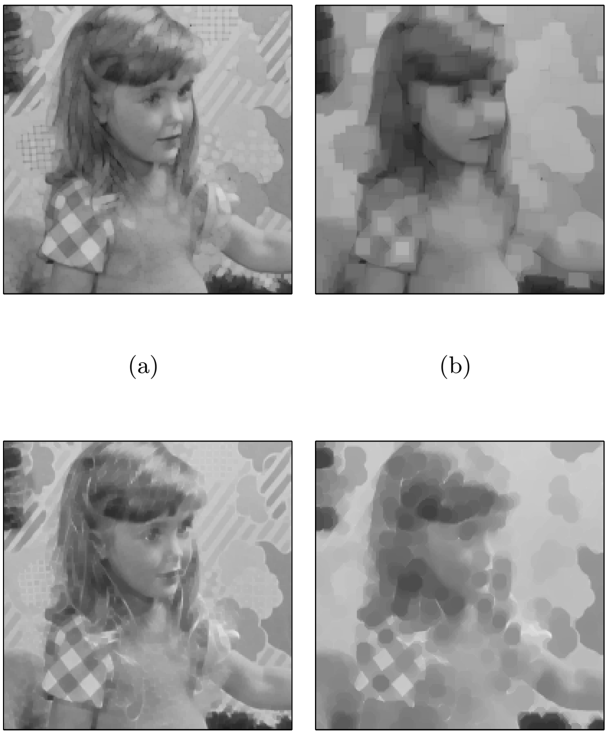
$$f \circ_{\mathrm{g}} k = (f \ominus_{\mathrm{g}} k) \oplus_{\mathrm{g}} k \tag{3.11}$$

Definition 14 (Grey scale closing) The grey scale closing of an image function f by a structuring element k is given by:

$$f \bullet_{\mathrm{g}} k = (f \oplus_{\mathrm{g}} k) \ominus_{\mathrm{g}} k \tag{3.12}$$

Thus opening is capable of removing regional maxima by first eroding the image and thus shrinking the maxima and then dilating the image so as to partially reconstruct features of the image which are not entirely removed by the erosion. In the initial erosion, maxima smaller than the structuring element are removed completely and as such are not reconstructed in the dilation. The larger the structuring element, the larger the maximum size of removed extrema. By duality, the closing of a grey scale image removes regional minima smaller than the structuring element. Figure 3.8 shows the section of the image of ‘girl’ given in Figure 3.7(a) after opening and closing with the same structuring elements as in the erosion and dilation examples from which the maxima and minima suppression is clear.

The morphological operators of erosion, dilation, opening and closing have been used extensively in many image processing applications including noise removal [12],



(a)

(b)

(c)

(d)

Figure 3.8: A section of the original image of ‘girl’ (a) along with the grey scale opening with a square structuring element of sizes 3 × 3 (b) and 9 × 9 (c) and the grey scale closing with a circular structuring element of radii 1 (d) and 4 (e).

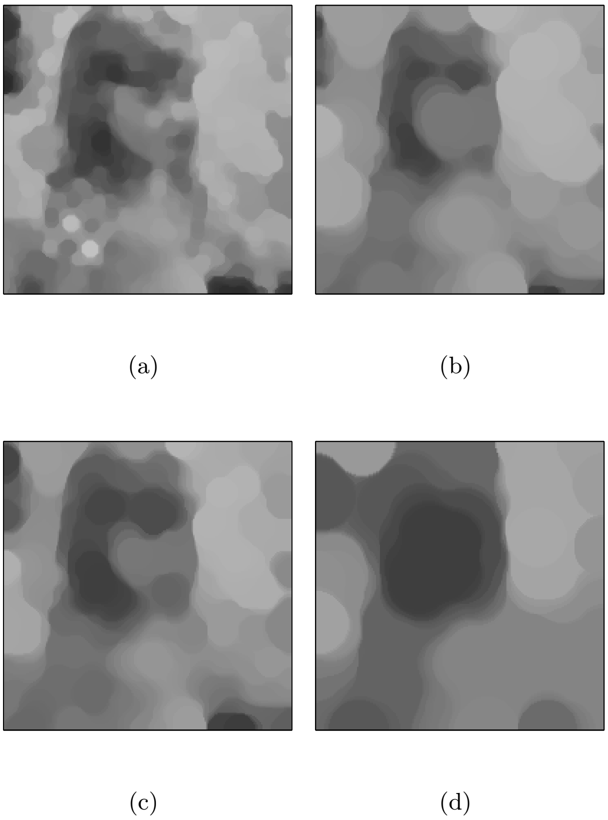


Figure 3.9: The section of the image of 'girl' presented in Figure 3.7(a) after open-closing with flat, circular structuring elements of radii 1 (a), 4 (b), 10 (c), and 20 (d).

multi-scale decomposition (sieves) and pattern recognition [27] and for the computation of the watershed transform [9]. By performing an opening followed by a closing, morphological operators are capable of removing both regional maxima and regional minima smaller than the structuring element used. Figure 3.9 shows the result of such an operation using flat, circular structuring elements of various sizes on the section of the image of 'girl'. By way of comparison, we include in Figure 3.10 the result of the linear filter with a flat, circular kernel of radius 10. Direct comparison of this image with that of Figure 3.9(c) shows that the morphological filter has not only further smoothed the image, but also to some extent preserved the edges of objects. This property of smoothing whilst preserving edges suggests that the morphological operation of open-closing (and indeed close-opening) is suited to image segmentation. Figure 3.11 shows the outline image of the open-closing of a

Figure 3.10: The result of performing a linear filtering operation the section of the image of 'girl' presented in Figure 3.7(a) using a flat, circular kernel of radius 10.

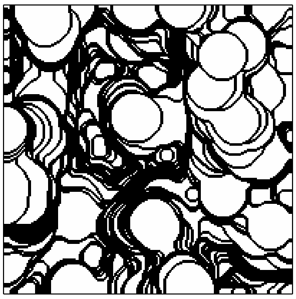


Figure 3.11: The outline of the open-closing of a section of the image 'girl' presented in Figure 3.9(b).

section of the image 'girl' presented in Figure 3.9(b). The outline image is taken to be 1 (represented here as black) where the gradient of the open-closing is non-zero and 0 (white) elsewhere. It is clear from the outline image that the boundaries of homogeneous regions of the open-closing do not correspond particularly well with objects in the image and there is a large dependence on the shape of the structuring element, the outline appearing to consist of overlapping circular regions the size and shape of which matches that of the the structuring element used. Though this tendency detracts from their useful application to image segmentation, the concept of suppressing regional minima and maxima in the image as a segmentation strategy is valuable.

Methods for overcoming the effect of the dependence of the result on the structuring element in grey scale morphology are referred to collectively as *adaptive* since the structuring element effectively changes as a function of spatial location. The literature reports that the concepts of adaptive morphological operators were developed under three separate titles; geodesic transforms [8, 9], attribute morphology [28] and filters by reconstruction [29]. Geodesic transforms operate under the constraint of a subset of the image domain. For example, geodesic dilation of the binary set A by the structuring element B under the set S, denoted $A \oplus_{gS} B$, for example dilates the foreground objects of A in the usual way provided the result does not expand beyond the boundary of the set S. Such geodesic transforms been shown to have applications in the determination of topographical image features, including the watersheds, and in particular, efficient algorithms have been developed for binary reconstruction. Filters by reconstruction are a direct consequence of geodesic transforms, which, along with attribute morphology and under the title connected operators, have been shown to be capable of image segmentation [30].

Attribute morphology as defined in [28] is based on the concept of a *maximum of openings* [31] in which an image is opened using several different structuring elements and the maximum of all results at each pixel location (effectively the grey level supremum) is taken to be the grey level of the result. The minimum of closings describes the dual operator. One classical attribute opening is that of area opening [28], the concept of which is that regional extrema below a given size threshold are suppressed in that they are given a grey level equal to the highest neighbouring pixel. For a size threshold of a, this is achieved by first opening the image with all possible structuring elements having a connected pixels and then taking the pixelwise maximum. The number of possible configurations for $a = 2$ is 4, for $a = 3$, 20 and the number for $a = 4$ runs into the hundreds. Clearly, exhaustive computation of the the openings is not possible. The authors of [28] propose an efficient algorithm for area opening / closing within the context of attributes, a later improvement (in terms of speed) is presented in [6]. The authors of [32] extend the use of attribute morphology in proposing an opening which uses not simply area as a criterion for merging, but any measure of a connected region. Recent work using attribute morphology includes that presented in [33] in which attribute morphology is used for the removal of noise in image sequences. Though approached from a slightly different theoretical background, attribute morphology produces results which can be achieved using connected operators and as such further discussion is entered into under this title.

The first grey scale filter by reconstruction to be presented in the literature is that of opening by reconstruction [29]. In the binary case, the image is eroded by a particular structuring element resulting in the shrinking of all, and complete removal of some foreground connected components. The remaining components are then dilated geodesically with respect to the original image to idempotence [8]. This *reconstruction* imposes the restriction that a connected component of the foreground set may be either completely retained or completely removed, but may not otherwise be altered. The result is that the shape of the structuring element determines only those features which are removed and does not appear in the opened image as in the grey scale case of Figure 3.11. The dual operator, closing by reconstruction can be calculated by first inverting the foreground and background and re-inverting after filtering. This has an identical effect on the image background as opening with the foreground. The technique for the extension to grey scale images is presented in the following section. The authors of [34] show that filters by reconstruction belong to the class of connected operators and as such work in both attribute morphology and filters by reconstruction has been developed more recently under this title [35]. It is to connected operators that we now turn our attention.

3.2 Connected Operators

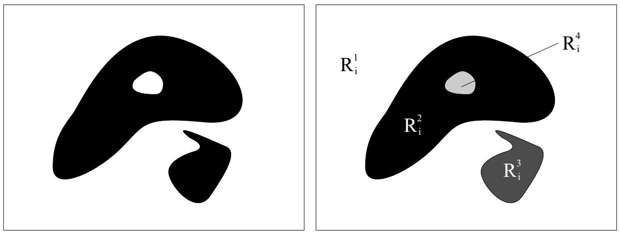
Connected operators form a set of operators in image processing that do not remove image components of specific frequencies like linear filters, they act upon the flat zones of the image. We begin this section by defining connected operators in the binary case. We then extend the definition to grey scale images by means of the definition of the partition of flat zones, discussed previously in terms of the partition lattice.

In order to define connected operators for binary images, we must first introduce some terminology. The fundamental principles of set theory, upon which the following relies, are presented in Appendix A. A digital, binary, still image can be represented by a mapping of two dimensional domain $D \in \mathbb{Z}^2$ onto the binary set $\{0, 1\}$. Thus, each value in the domain of the image is permitted to take a value of 0 or 1. The set of points (pixels) in the domain taking a value of 1 are often referred to as the foreground and the set of pixels taking a value of zero as background. The complementation of a binary image can now be defined as the swapping foreground for background or alternatively performing a logical NOT operation on each pixel. The set difference operator $\setminus$, returns a binary image taking a value of 1 where the two operand images are different. Finally we introduce the concept of a pair of connected pixels and a connected component. When using a square grid for the domain of the image there are two basic connectivities which can be observed, referred to as four- and eight-connectivity. In four connectivity, pixels to the left, right up and down are considered connected, whilst those on the diagonals are not. Eight-connectivity permits pixels in the diagonal neighbourhood to be considered connected. Thus, two foreground (background) pixels are said to be connected if there can be found between them a path of connected foreground (background) pixels. This leads to the concept of a connected component, being a group of foreground or background pixels which are all connected.

The definition of a connected operator for binary image sets can now be defined as follows.

Definition 15 (Connected Operators for Binary Image Sets) A binary set operator ψ is said to be connected if for any binary image set A, the set difference $A \setminus \psi(A)$ consists exclusively of connected components of A or its complement A^c.

That is to say that the operator serves only to remove or preserve connected components of either the foreground or the background. It is clear from this definition that binary set operation performed on each level set in opening by reconstruction is a connected operator since it can only preserve exactly or remove completely connected regions associated with the foreground of the image. In its current form, this definition cannot be extended to grey scale image functions directly, we must first define binary connected operators in terms of the lattice or partitions. A partition $\mathcal{P}_i$ being a set of regions (often referred to as *partition classes*), $\{\mathcal{R}_i^n\}$, obeying Definition 1 is created such that each region defines a connected component of either the foreground or the background of the image. Figure 3.12(a) shows a simple binary image a possible partition of which is given in Figure 3.12(b). Returning to the ordering relation used to define the lattice of partitions, a partition $\mathcal{P}_i$ is said to be finer than another $\mathcal{P}_j$ if all points belonging to one partition class $R_i^n \in \mathcal{P}_i$ belong



(a) (b)

Figure 3.12: A simple binary image (a) and a possible partition thereof (b).

to one and only one partition class of $\mathcal{P}_j$, $R_j^m \in \mathcal{P}_j$. We may now define binary connected operators in terms of the ordering of the partitions of operand and result.

Definition 16 (Connected Operators in Terms of Partitions) A grey scale operator, ψ, is said to be connected if, when acting on a set, A, the partition associated with $\psi(A)$ is less fine than that associated with A.

Thus for binary images, the 'fineness' of the partition associated with the result can only be less than that associated with the original image if partitions are either completely removed or completely preserved. This can be extended to grey scale images by choosing a suitable method of partition. The authors of [34] show that the *partition of flat zones* constitutes one such method of partition, flat zones being the largest connected areas of the image function for which the image grey level is constant. The implication of this choice of partition is that connected operators operate on only a subset of the lattice of partitions. We may now define connected operators for grey scale images.

Definition 17 (Connected Operators for Grey Scale Functions) An operator, Ψ , acting on a grey scale function, f, is said to be connected if the partition of flat zones associated with $\Psi(f)$ is less fine that the partition of flat zones associated with f.

The effect of connected operators on grey scale images has been described previously in terms of the merging of homogeneous regions, the above definition formalises this notion. Considerable attention has been paid to the design of new connected opera-

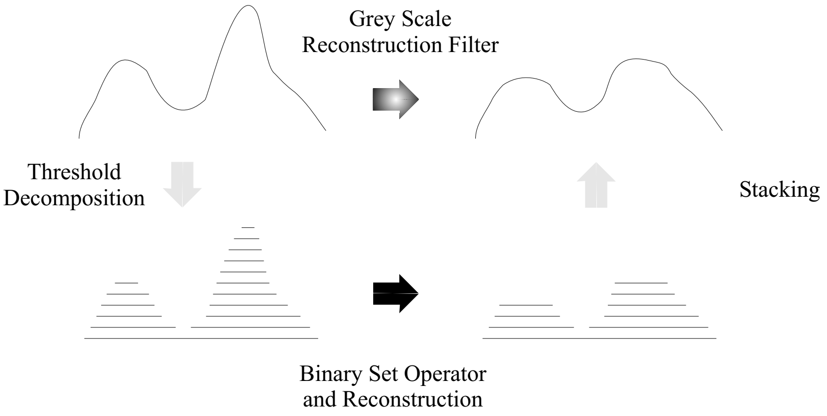


Figure 3.13: A schematic of the stages of a grey scale filter by reconstruction, decomposition, binary filter / reconstruction and stacking.

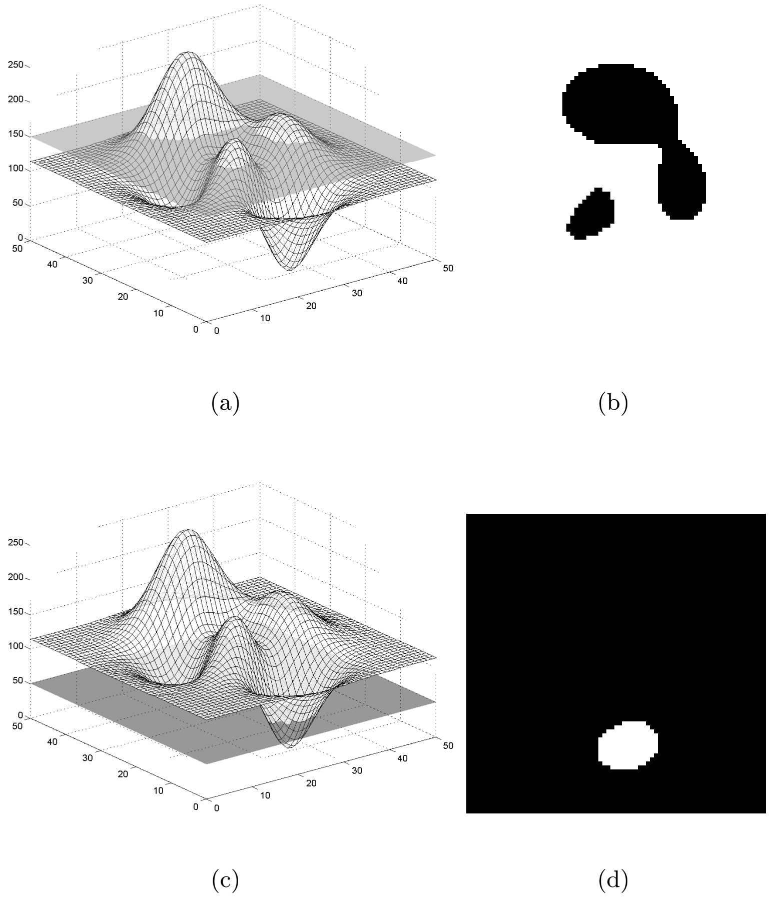
tors, we shall discuss a number of these shortly, however we shall firstly concentrate on the extension of binary connected operators to operate on grey scale images.

Figure 3.13 describes the arrangement by which any binary connected operator may be extended to act upon grey scale images. It consists of three distinct stages; decomposition, filtering and stacking. In the first, the image with N grey levels is decomposed as a set of N binary images, each of which, $S_{\mu=1,2,\dots N-1}$, takes the value 1 at pixel locations where the grey level of the original image is higher than μ and 0 elsewhere. Formally, each *level set* of an image $f(\mathbf{x})$ is given by the following:

$$S_\mu(\mathbf{x}) = \begin{cases} 1 & \forall f(\mathbf{x}) \geq \mu \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

Figure 3.14 shows the image ‘peaks’ with superimposed threshold levels of 150 and 50 along with the corresponding level sets, black representing the value 1 and white, 0. Note that the surface representation has been rotated relative to the image in order to allow the minor peaks to be observed. This convention has been adopted whenever the image of ‘peaks’ is represented as a surface plot.

In the second stage, each level set is operated on by the binary connected operator $\psi(S_\mu)$. In the case of opening by reconstruction the operator ψ involves the erosion and reconstruction procedure described previously. In general though, we may wish to analyse each foreground connected component of the level set separately. The



(a)

(b)

(c)

(d)

Figure 3.14: The image of 'peaks' with superimposed threshold level and the corresponding level sets for values of 150, (a) and (b) and 50, (c) and (d).

connected components of the level set S_μ can be defined as follows:

$$C_{S_\mu}(\mathbf{p}) = \{\mathbf{q} \mid P_{S_\mu}(\mathbf{p}, \mathbf{q})\} \tag{3.14}$$

Where $P_{S_\mu}(\mathbf{p}, \mathbf{q})$ is an unbroken path between points $\mathbf{p}$ and $\mathbf{q}$ consisting of foreground pixels of the level set S_μ. Thus associated with each point of the image $\mathbf{p}$ is a set of pixels $C_{S_\mu}(\mathbf{p})$ which belong to the same connected component of S_μ as $\mathbf{p}$. Each foreground pixel of the binary image set can now be preserved or removed based on some measure of the connected component to which it belongs. We shall discuss such measures shortly.

Once all level sets have been operated on, the complete set of filtered binary images are stacked such that the grey level of the result is given by the maximum value μ for which the level set S_μ is non-zero. This can be formalised by the supremum operation:

$$\Psi(f) = \bigvee_l \left(\bigcap_{\mu<l} \psi(S_\mu) \right) \tag{3.15}$$

It should be noted that the opening by reconstruction used thus far as illustration of these methods is such that if a connected component of the level set S_μ is removed then all connected components of level sets associated with higher grey levels $\eta > \mu$ which 'hit' the connected component are also necessarily removed. The term 'hit' implies that the intersection of the two connected components does not result in the empty set. Such operators are termed increasing.

Definition 18 A connected operator acting on a grey scale image is increasing if and only if the following inclusion relation holds.

$$\psi(S_\eta) \subseteq \psi(S_\mu)\ \forall\ \eta \geq \mu \tag{3.16}$$

This is not to be confused with increasing in terms of the lattice of partitions and requires in most cases that the function from which the levels sets are derived is single valued, as is the case for image functions. This method describes the application of binary opening operators to grey scale images. Grey scale closings can be calculated either by using the corresponding closing binary connected operator in the filtering phase, or by inverting the image prior to performing the opening as described, the effect is identical.

This method is described for generalised connected operators in [29, 30] and [36] in which the authors refer to three classical connected operators, opening by reconstruction, area opening and the λ-max operator. Also proposed in [36] is the complexity operator. Having already discussed opening by reconstruction, we shall now introduce the operators of area opening, λ-max and the complexity criteria connected operator.

In the binary case, area opening removes from the foreground any connected components with an area, in pixels, less than a given threshold. In the application of this operator to the level sets of a grey scale image, the decision to retain a pixel $\mathbf{p}$ in the foreground level set is made on the basis that the area of the connected component $|C_{S_\mu}|$, given in practice by the length of the set of connected pixels, is less than the threshold.

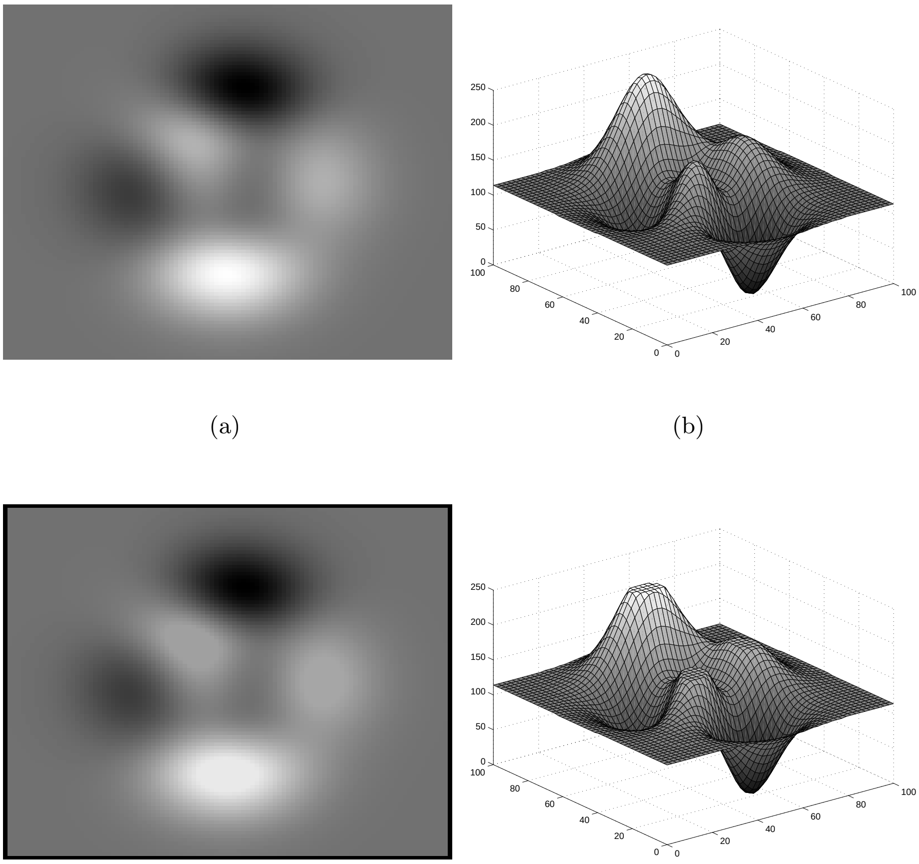
$$S_\mu \circ_a (a) = \left\{\mathbf{p} \mid |C_{S_\mu}(\mathbf{p})| < a\right\} \tag{3.17}$$

Note that the area opening operator is increasing both in terms of the level set ordering (Definition 18) and in terms of the lattice of partitions since it is order preserving. Also, in terms of the lattice of partitions, it is idempotent and anti-extensive, making it a morphological operator. The effect of performing an area opening on a grey scale image is to truncate peaks of the image such that no regional maxima exist with an area less than the area threshold a. Figures 3.15(a) and 3.15(b) show the original image of 'peaks' and a representative surface plot respectively. Corresponding plots for the image of 'peaks' after area opening with an area threshold of 100 pixels are given in Figures 3.15(c) and 3.15(d). The dual operator, area closing removes from the binary image or level sets of a grey scale image connected components of the background of size less than a.

$$S_\mu \bullet_a (a) = \left\{\mathbf{p} \mid |C_{S_\mu^c}(\mathbf{p})| < a\right\} \tag{3.18}$$

Where S_μ^c is the complement of the level set S_μ. The literature reports that area openings and closing (often collectively termed area operators) have been used extensively in such areas as edge detection [37], scale space classification [38] and image simplification for compression [39]. Appendix B demonstrates a novel use of area operators in the removal of impulsive noise from images, and in the next chapter, we investigate the use of area open-closing for image segmentation.

In [36], the λ-max operator is presented as an increasing connected operator in which



(a) (b)

(c) (d)

Figure 3.15: The image of ‘peaks’ (a), a surface representation thereof (b) and the corresponding visualisations for the image after area opening with an area threshold of 100 (c) and (d).

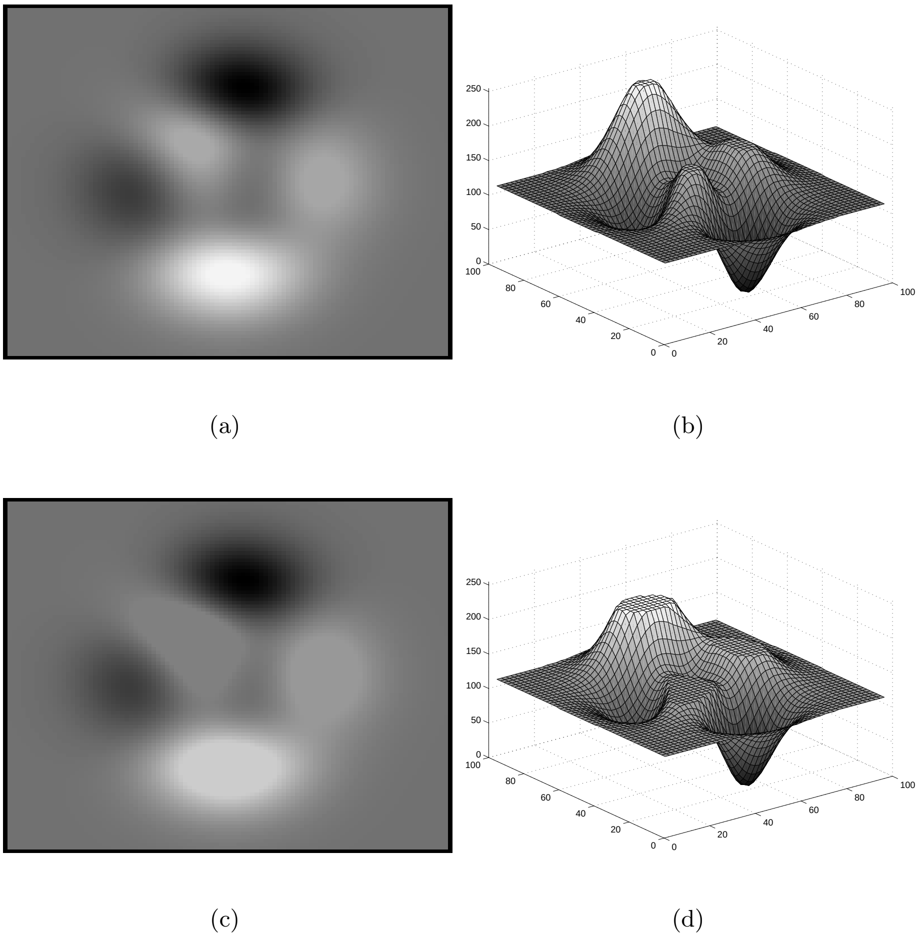


Figure 3.16: The image of 'peaks' after application of the λ-max operator with $\lambda = 10$ (a), a surface representation thereof (b) and $\lambda = 50$ (c) and (d).

connected components of a level set S_μ are removed if their intersection with the level set $S_{\mu+\lambda}$ results in the empty set. This can be formalised thus:

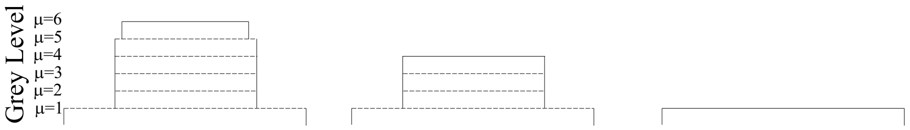
$$S_\mu \circ_{\lambda-\max} (\lambda) = \big\{\mathbf{p} \mid C_{S_\mu}(\mathbf{p}) \cap S_{\mu+\lambda} \neq \emptyset\big\} \tag{3.19}$$

The result is the reduction in grey level of these maxima by λ grey levels. Figure 3.16 presents images and surface plots of the image 'peaks', the original of which is presented in Figure 3.15(a) after application of the λ-max operator for $\lambda = 10$ and $\lambda = 50$. Comparison of the surface plots with those presented in Figure 3.15 demonstrates the difference between the two operators. Whilst both have truncated the peaks (bright regions) of the image, for the area opening, the area of each remaining plateau is equal to the parameter a, while for the the λ-max operator, each peak has been reduced by at most λ grey levels.The λ-min operator describes the dual operator in which regional minima of low contrast are removed.

$$S_\mu \bullet_{\lambda-\min}(\lambda) = \left\{ \mathbf{p} \mid C_{S_\mu^c}(\mathbf{p}) \cap S_{\mu+\lambda}^c \neq \emptyset \right\} \tag{3.20}$$

However in a later paper [40], the authors describe a λ-max operator in terms of the *max-tree*, a convenient framework for the design and implementation of connected operators which differs subtly from the above definition. Without entering into a full discussion of the max-tree representation, the operator removes from the binary level set S_μ any connected component whose intersection with the level set $S_{\mu+\lambda}$ results in the empty set, as in the previous definition. The operator differs in its treatment of connected components of the level sets below μ in that any connected component of the level set $S_{\mu-h},\ h = 1, 2...\mu$ with which the union of the removed component C_{S_μ} results in the connected component C_{S_μ} is also removed. The effect of this operator, which we shall refer to as the adaptive λ-max operator is best illustrated by a simple one-dimensional example. Figure 3.17(a) shows the level sets of a simple one-dimensional signal. Figure 3.17(b) shows the result of the λ-max connected operator for $\lambda = 2$, note that only the connected components of the top two level sets have been removed. Figure 3.17(c) shows the result of the adaptive λ-max operator for $\lambda = 2$ in which not only have the connected components of the level set for $\mu = 6$ and 7 been removed, but also those connected components below $\mu = 6$ for which the union with with the component removed at $\mu = 6$ results in said component. That is, whereas the λ-max operator for $\lambda = 1$ reduces by one grey level the height of regional maxima, the adaptive λ-max operator can be said to remove completely regional maxima, giving the associated pixels a grey level equal to the highest grey level of its neighbours. Thus, the value of λ changes as a function of the image contrast, hence we refer to it as adaptive. This subtle, but significant difference implies that the adaptive λ-max operator is best implemented with an iterative algorithm, the value of λ describing the number of required iterations. By duality, the adaptive λ-min operator can be defined. These operators are compared with area and standard morphological operators for the removal of impulsive noise and also, in the next chapter image segmentation.

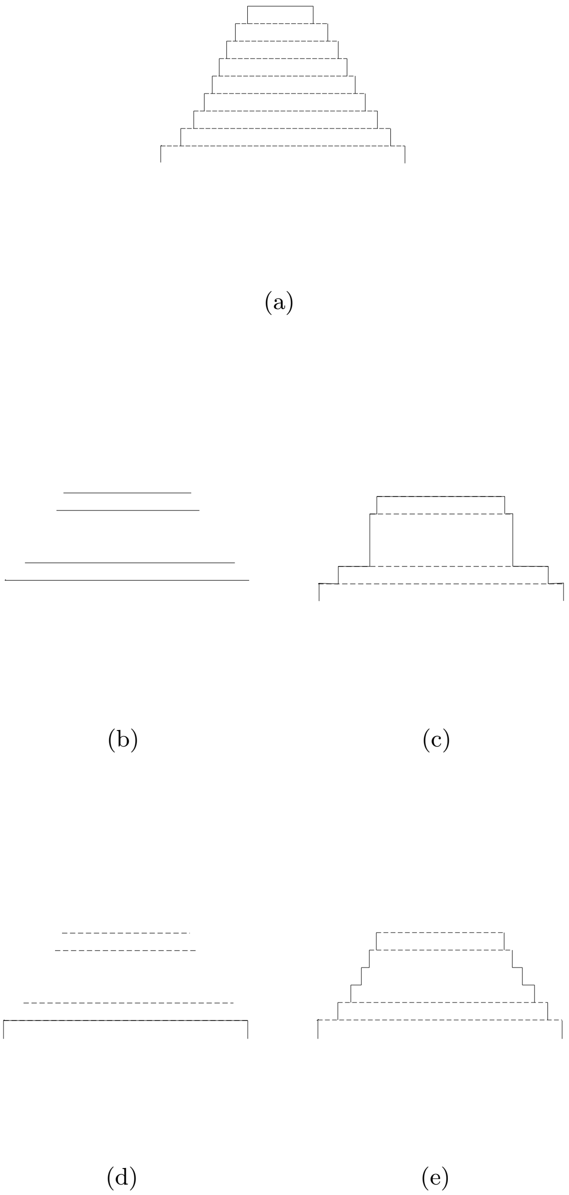
The complexity operator, as the name suggests, removes from the level set any connected component having a complexity greater than a given threshold. The authors of [36] propose a complexity measure calculated as the ratio of the connected com-



(a) Original. (b) λ-max. (c) Adaptive λ-max.

Figure 3.17: The level set decomposition of a simple one-dimensional signal (a) and the resulting level sets after application of the λ-max (b) and adaptive λ-max (c) connected operators.

ponents perimeter to its area, $\mathcal{C} = \left\langle C_{S_\mu} \right\rangle / |(C_{S_\mu}|$ where the area, $|\ |$ is calculated as before and the perimeter, $\langle\ \rangle$ is the number of pixels in the connected component that neighbour a background pixel of the level set. Unlike the area and λ-max operators, the complexity of a connected component of the level set S_μ being high, resulting in its removal, does not necessarily imply that a connected component of a higher level set which hits said connected component will also be removed. The complexity operator is therefore non-increasing. The use of non-increasing criteria for connected component removal raises issues with regard to the stacking process i.e. if a connected component of the level set S_μ is removed, what should become of all the connected components of higher level sets which hit the removed component. This problem, along with three simple solutions as presented in [40] are given for a one dimensional signal in Figure 3.18. The original signal is shown in Figure 3.18(a) with the connected components of each level set shown with dashed lines. The connected components remaining after filtering each level set with some non-increasing operator are shown in figure 3.18(b). Figure 3.18(c) gives the resulting signal using what is termed a direct decision which follows that according to Equation 3.15. The result is a signal with large jumps in grey level. Figure 3.18(d) shows the result of a minimum decision in which connected components above the lowest removed are ignored in the stacking procedure. Finally, the maximum decision, in which the original signal is preserved up to the maximum retained connected component is shown in Figure 3.18(e). One further method for making the decision to preserve or



(a)

(b)

(c)

(d)

(e)

Figure 3.18: A simple one dimensional signal with the binary connected components superimposed (a), the level sets retained after filtering (b) and the signals resulting from a direct decision (c), a minimum decision (d) and maximum decision (e).

remove a connected component presented in [40] involves a viterbi optimisation and is shown to have desirable results. Recent advances by the authors of [36, 41–43] involve the use of the max-tree, in the context of which they develop a number of interesting connected operators two of which, an operator based on entropy and another on motion, we shall briefly discuss here.

For both the entropy and motion operators, the decision to retain a connected component of a level set is not based solely on the properties of that binary connected component, but on the section of the image under its region of support. In the case of the entropy operator [40], the grey scale probability density function $P_{C_{S_\mu}}(\mu)$ is estimated by normalising the grey scale histogram from which the entropy is estimated according to:

$$H(C_{S_\mu}) = -\sum_{\mu} P_{C_{S_\mu}}(\mu) \log_2[P_{C_{S_\mu}}(\mu)] \tag{3.21}$$

The entropy of the region can be thought of as an indication of the amount of information in that region. Regions of low entropy can therefore be removed without greatly effecting the information content of the image.

The motion operator first proposed in [42] and developed in [43] uses the motion of image regions between consecutive frames of an image sequence to assess the inclusion criterion. The operator applies to the area of the image under the region of support of the connected component a specified motion model. If, after application of this model the difference between the translated block and the next frame of the image sequence falls below a specified threshold it can be assumed that that region obeys the applied motion model and the associated connected component is removed from the level set. Thus, regions of the image obeying the specified motion model are removed form the image. The entropy and motion based operators have been included as examples of connected operators that have not been extended from binary operators. In the sequel, we investigate a novel connected operator which not only uses grey level information, but also does away with the level set formulation in that it merges homogeneous regions based on their individual properties.

3.3 Summary

In this chapter, we began by introducing the lattice of partitions, a framework under which the region based techniques shown to have potential in the previous chapter could be unified. Morphological operators are then defined for both binary and grey scale images and examples are given which demonstrate their property of image smoothing whilst preserving edges. It is demonstrated that whilst this property appears to make such operators well suited to image segmentation, the homogeneous regions of the resulting images have a shape which is highly dependent upon the shape of the structuring element used. Three adaptive methods which overcome the problem are then presented of which attribute morphology and filters by reconstruction are further developed under the title connected operators. Examples of classical connected operators including area and λ-max operators are discussed and in Appendix B shown to have application in the removal of impulsive noise from grey scale images. In the following chapter, we develop the classical connected operators for image segmentation and show that they can be used to produce novel segmentation methods which produce encouraging results.

Chapter 4

Connected Operators for Partial Image Segmentation

In the previous chapter, connected operators were introduced and several examples including the area, λ-max and adaptive λ-max open-closing operators were shown to have application in the removal of salt and pepper noise from grey scale images. In this chapter, we first introduce a novel metric for the objective evaluation of a partial segmentation based on the coding cost and then, using this metric demonstrate that connected operators can be used as a segmentation tool but suffer from over-segmentation of regions we shall refer to as transition zones. We present two novel connected operators which act to remove these transition zones and improve the result of the segmentation in the context of the proposed metric. We then present an adaptation of the metric which permits some control over the degree of segmentation achieved. Finally we present one further novel operator which is designed for the minimisation of the metric and present some results.

4.1 A New Metric for the Objective Evaluation of a Partial Segmentation

A very important question in the field of partial image segmentation is how the success of the result can be quantified. The ability to attach to a segmentation a measure of 'goodness' allows parameters of a segmentation, such as the area threshold or λ in the example of connected operators, to be optimised. It also allows segmentations produced using different algorithms to be compared quantitatively. Several authors [44–48] use a hand-segmented 'ground truth' image in which the outlines of objects or significant regions in the image are given by human inspection, while others use the human psychovisual evaluation of the resulting segmentation [49]. Such methods of segmentation evaluation have a highly subjective element in the human inspection and, in any case, base their evaluation on the location of segment boundaries disregarding the information inside segments. The method due to [50] differs from these in that it is apparently an objective measure using the expression given in Equation 4.1 to evaluate the segmentation $\mathcal{P}_f$ of an image function $f(\mathbf{x})$ of dimensions $M \times N$ in which I is the number of segments, A_i and e_i are the area and average colour error of segment $\mathcal{R}_i$.

$$\Pi(\mathcal{P}_f) = \frac{1}{1000(M \times N)} \sqrt{I} \sum_{i=1}^{I} \frac{e_i^2}{\sqrt{A_i}} \tag{4.1}$$

The lower the measure $\Pi(\mathcal{P}_f)$, the better the segmentation. Though objective in its calculation, this measure is empirical in that is has been established as a measure which is found to give subjectively pleasing results. A second shortcoming is that as the error between image content and segment value approaches zero, the $\frac{e_i^2}{\sqrt{A_i}}$ term decreases more rapidly than the $\sqrt{I}$ term increases and as such the algorithm favours segmentations with larger numbers of regions, a problem highlighted in [51]. The authors of [51] present an adaptation of this metric which attempts to overcome this, but in either case the absence of a tuning parameter suggests that the segmentation for which Π is a minimum is 'the best' and does not take into account the fact that, for example, more, or less regions may be required depending on the scene content and the application. This may seem to lead to a rather circular argument as if there is a parameter in the evaluation metric, then we must choose this parameter, and the very reason for the metric is to remove any subjective appraisal of the segmentation. This stalemate can be mitigated by considering that the parameter for the metric need only be selected once and thereafter segmentations evaluated using the metric may be compared using a consistent measure.

Having highlighted the desirable properties of a metric for segmentation evaluation, we must identify the desirable properties of a segmentation upon which a metric should be based. The criteria we present here are very general and not specific to one single application, such as motion estimation, and therefore differ from those given in Chapter 2. One established set of such criteria is due to [2] which we recapitulate here.

- Segments should be uniform and homogeneous with respect to some characteristic such as grey level or texture.
- Segment interiors should be simple and without many small holes.
- Neighbouring segments should be significantly different with respect to the characteristic on which they are uniform.
- Segment boundaries must be simple and spatially accurate.

With these in mind, we present a novel metric for the evaluation of an image segmentation based upon the cost of coding the image using the mean replaced segmentation image (in which each segment is filled with the mean grey level of the image under its support) as an approximation to the original. It should be noted that the presented metric applies equally well to segmentations of colour images, but for consistency we restrict our study to grey scale images. The underlying principle of the metric is that the lower the cost of coding the image using the segmentation the better the segmentation. This has analogies with the minimum description length principle in which one seeks the shortest model required to describe a stationary, random data set [52, 53]. It will be demonstrated that the coding cost is related to the number and location of segments, the fundamental properties of any segmentation, and in a later section, that the introduction of a single weighting parameter allows the metric to favour segmentations with particular properties as the scene content and application may require.

The coding cost is evaluated through a simple coding scheme analysing individually three components taken from the original image and the segmentation. These comprise the following;

- The outline of the segmentation; being an image describing the location of segment boundaries.
- The set of values describing the mean grey level of the original image inside each of the segments.
- The error image; taking for each pixel in the image domain a value equal to the difference between the mean of the segment in which that pixel lies and the value of that pixel in the original image.

From these three components we can perfectly reconstruct the image by applying the mean values to the regions defined by the outline and adding to this the error image. This underlines the fact that the implied coding scheme is lossless. The theoretical minimum cost of the lossless coding of the error image and the segment means is calculated via the entropy and expressed in bits per pixel. The theoretical limit imposed by the entropy does not apply to the outline image as there is a significant degree of spatial redundancy, i.e. neighbouring pixels are likely to have similar values. As such we propose a simple scheme which exploits to some extent this redundancy and show that using this scheme we can achieve results lower than the raw entropy predicts. The total cost of coding the image using the segmentation is given by the sum of the coding costs of the three components.

4.1.1 The Segmentation Outline

We choose to present the outline of a segmentation by an image containing four levels, (0 to 3). These values are determined in raster scan fashion. A value of 0 indicates that the pixels to the right and below are of the same segment, 1 indicates that the pixel to the right belongs to a different segment but the pixel below is of the same segment, 2 indicates that the pixel below is of a different segment but the pixel to the right is of the same segment and 3 indicates that both the pixel below and the pixel to the right are of different segments. Pixels lying on the right hand and lower edges of the image are given values which assume that edge of the image represents a segment boundary. It will be seen that this ensures consistency between the sum of coding costs of all three components and the entropy of the original image. The result of the allocation (I_o) is then a complete description of the segmentation outline having the same dimensions as the original image which we shall denote $M \times N$.

The spatial redundancy in this full image is reduced by calculating a single 8-bit value derived from each 2×2 block of I_o. The result is an image I_{or} having dimensions $\frac{M}{2} \times \frac{N}{2}$. The following equation describes this operation.

$$\begin{aligned} I_{or}(i,j) &= I_o(2i,2j) + 4I_o(2i+1,2j) + 16I_o(2i,2j+1) + \ldots \\ &\quad \ldots 64I_o(2i+1,2j+1),\ 0 < i < \frac{M}{2},\ 0 < j < \frac{N}{2} \end{aligned} \tag{4.2}$$

The values of the compressed outline image, which lie in the range $0 < I_{or}(i,j) < 255$, are unique for any particular combination of pixels in the outline image I_o and as such the uncompressed outline image can be perfectly reconstructed. It should be noted that the compression operation assumes that the image dimensions are even values although images with odd dimensions can be accommodated with a small loss of efficiency. The entropy of the compressed outline image H_{or} is calculated according to Equation 4.3 where $p_{or}(k)$ is the probability of a pixel having a value value $0 < k < 256$ and is derived from the normalised histogram.

$$H_{or} = -\sum_{0}^{255} p_{or}(k) \log_2 p_{or}(k) \tag{4.3}$$

The outline entropy is high if there are an equal number of each pixel value in the outline image, such as is the case in a segmentation with very complex boundaries and low if all pixels have the same value. In the limit as each segment represents a single pixel, the entropy of the outline image is zero (considering that the edges have been regarded as boundaries). In the opposite extreme of a single segment, the outline entropy approaches zero, but does not reach it on account of the pixel values at the right hand and lower edges.

In order to express the outline entropy in terms of the minimum number of bits required to represent each pixel of the original image, we introduce the outline coding cost C_{or}, such that

$$C_{or} = \frac{1}{4} H_{or} \tag{4.4}$$

A similar coding cost C_o can be derived for the interim outline image having a

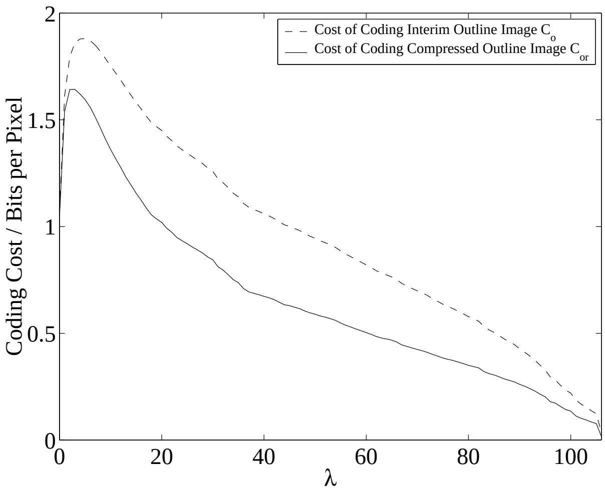


Figure 4.1: The coding cost of the interim outline image C_o and the outline image exploiting spatial redundancy C_{or} for the image of 'girl' after segmentation using the adaptive λ-max operator as a function of λ.

probability distribution $p_o(0 < k < 3)$.

$$C_o = H_o = -\sum_{0}^{3} p_o(k) \log_2 p_o(k) \tag{4.5}$$

Note the absence of a normalisation factor due to the fact that the interim outline image has the same dimensions as the original image and as such the entropy already has the units of bits per pixel. Figure 4.1 shows the coding costs C_o and C_{or}, for the segmentation outline of the image of 'girl' having been segmented using the adaptive λ-max open-closing operator as a function of λ. The results clearly show that the coding cost of the outline image after exploitation of the spatial redundancy C_{or} is significantly lower than the coding cost of the interim outline image C_o. It is suggested that even more efficient coding strategies such as run length encoding may be employed which further exploit spatial redundancy to reduce this cost, however a full study is not entered into here.

The cost of coding the outline of the segmentation in this way leads to a measure which is high if the segment outlines are very complex and lower for simple outlines. Indeed, the outline coding cost is zero when segments represent single pixels, but as Figure 4.1 demonstrates it rises rapidly with increasing λ as segment boundaries become more complex reaching a peak while $\lambda < 10$. As segments become larger and the boundaries become less complex, the outline coding cost falls steadily, approaching zero when the segmentation has a single segment ($\lambda = 128$). Thus this component of our metric directly addresses the property of having simple boundaries described in the introduction to this chapter. The outline coding cost is also instrumental in addressing the criteria of having no similar neighbouring segments and many holes inside regions. After the initial rise (demonstrated in Figure 4.1), the majority of pixels lie inside segments and as such have the same outline image pixel value. This leads to a peak in the probability distribution and hence a lower entropy. The entropy can be reduced further by assigning segment boundary pixels to segment interiors (merging regions) such that the outliers of the distribution join the dominant mode. Therefore, merging regions reduces the outline entropy, be these holes or neighbouring segments with similar mean grey levels.

4.1.2 Segment Mean Values

Each segment of the segmentation has associated with it a set of pixels $\mathcal{R}_i$. In calculating the mean replaced segmentation image the pixels in each region are given a grey level equal to the mean grey value of the associated pixels in the original image. The entropy of these mean values for an eight bit grey scale image, is calculated according to Equation 4.6 where $0 \leq m_k < 256$ are the grey level values and $p(m_k)$ is the probability of a segment having an associated mean value of m_k, i.e. the number of segments having that mean grey level. In many cases this will be a rounded mean as grey levels must take integer values in order for the discrete entropy calculation to be applied but this does not inhibit the metric as the error image, which must also consist of discrete values is calculated according to this rounded mean. The result H_m, is the minimum number of bits per segment required to code the mean values.

$$H_m = -\sum_{m_k=0}^{255} p(m_k) \log_2 p(m_k) \tag{4.6}$$

As for the outline component, we wish to express the cost of coding the segment means in terms of a number of bits per pixel of the original and as such, the means coding cost C_m is defined as

$$C_m = \frac{N_s}{MN} H_m \tag{4.7}$$

Where N_s is the number of segments and $M \times N$ is the dimension of the original image. The factor $\frac{N_s}{MN}$ indicates that the cost of coding the segment means will be high for large numbers of segments (relative to the number of pixels) and so to some extent this component of the total coding cost penalises segmentations with many segments. The entropy H_m will also be highest for the pixel segment segmentation as the probability distribution is likely to be relatively flat. As the image is segmented such that segments become larger and fewer, the probability distribution will tend to cluster around particular values and the entropy will drop. In this way, a segmentation having two neighbouring segments with very similar grey levels will have not only a higher outline coding cost as mentioned in the previous section, but also a higher cost of coding the means, and thus the segmentation will necessarily have a higher total coding cost than one in which the two regions were represented by a single segment. If the coding cost of the means is calculated for the segmentation in which each segment represents a single pixel, the entropy of the mean values is identical to that of the original image and for a segmentation with only one segment, the means coding cost is zero.

The coding cost of the means therefore addresses the criteria of segment interiors having no small holes and neighbouring segments having significantly different mean grey levels.

4.1.3 Error Coding Cost

The error image is calculated by subtracting the grey level of every pixel from the rounded mean of the segment in which that pixel lies and as such, in the case of an 8-bit image, can take any value from -255 to 255. The cost of coding the error is equal to the entropy of this error image which is calculated according to Equation 4.8, in which e_k are the set of all error values present in the error image, and $p(e_k)$ is the probability of a pixel in the error image having a value of $-255 \leq e_k \leq 255$ derived once again from the normalised histogram.

$$C_e = H_e = -\sum_{e_k=-255}^{255} p(e_k) \log_2 p(e_k) \tag{4.8}$$

The error coding cost is thus high if regions are not homogeneous in which case the error image has a relatively broad and flat probability distribution. A distribution

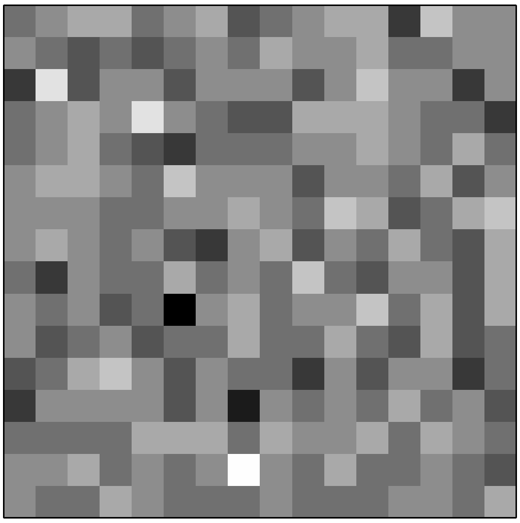


Figure 4.2: A 256 × 256 pixel image of 16 × 16 pixel blocks each having a random grey level.

in which error values are clustered around zero, indicating that regions are largely homogeneous, will have a considerably lower coding cost and in the limit as segments represent single pixels, the error for each pixel and therefore the error coding cost are zero. Thus the error coding cost addresses the last criteria of region homogeneity.

The total cost of coding the image, expressed in bits per pixel is thus given by the following.

$$C_{tot} = C_{or} + C_m + C_e \tag{4.9}$$

To illustrate this coding scheme, we apply it to a carefully selected example. Figure 4.2 shows a 256 × 256 pixel image which has been split into 16 × 16 blocks, each block being assigned a random, 8-bit grey level based on a Gaussian distribution, the entropy of which without segmentation is evaluated to be 2.33 bits per pixel.

The component and total coding costs are then calculated for a set of segmentations, each consisting of regular $m \times m$ pixel blocks where $m = 2^{1,2,\ldots,8}$. Figure 4.3 shows the coding costs as a function of block size m. For a segment size of one pixel, we see that the outline coding cost is zero due to the fact that all pixels of the outline image have been assigned a value of 3. The error coding cost is also zero since the mean value for each segment is the pixel value of the original, a status which

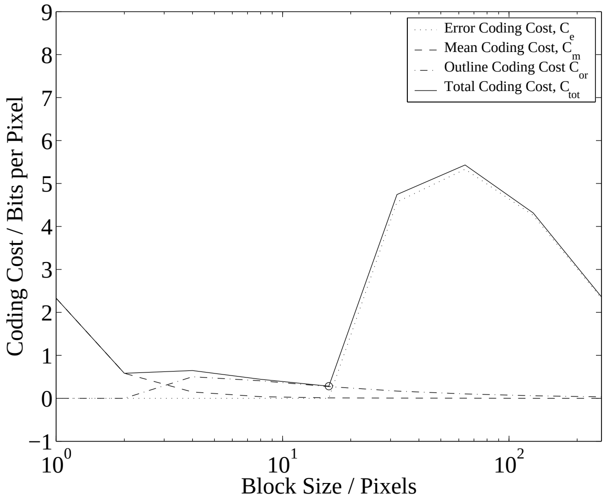


Figure 4.3: The component and total coding costs of the segmented block image as a function of segment size (m).

continues with increasing segment size until the size is greater than the block size of the original image. The sole contribution to the total coding cost for single pixel segments is the coding cost of the means which in fact equals the entropy of the original image.

Increasing the block size has the initial effect of increasing the outline coding cost as the probability distribution of I_{or} becomes broader. This reaches a maximum at a block size of 4, after which it decreases steadily until there is a single segment for which it approaches zero since the vast majority of pixels of the outline image (with the exception of those at the right hand and lower edge) have the same value, 0. The means coding cost however falls with increasing segment size due to the decreasing number of segments N_s for $m < 16$ and also for the change in the probability distribution for $m > 16$. For a single segment ($m = 256$) there is only one mean value and as such the means coding cost drops to zero. The error coding cost increases sharply after the segment size exceeds that of the blocks in the original image, the result is an increase in the hitherto decreasing total coding cost creating a minimum when segment size corresponds exactly with the block size, indicating that the optimum block size for this segmentation scheme is 16 as one might expect.

In the following sections we will use the coding cost metric presented above to optimise the parameters of various connected operators. The introduction of a weighting parameter which allows the metric to be tuned to suit a specific image or application is presented after demonstrating the un-weighted metric on real images segmented by the application of connected operators.

4.2 Area λ-max and adaptive λ-max Open-closing

This section demonstrates that the property of removing regional extrema whilst preserving important shape information makes connected operators suitable for image segmentation. We begin by selecting the parameter of the connected operator, a or λ for which the metric is lowest and demonstrate that these segmentations comprise two types of zone; flat zones, which correspond well with the interiors of significant regions, and transition zones which prevent segment boundaries corresponding well with the boundaries of significant regions.

When applying connected operators to the segmentation of images, segments are taken to be regions for which the grey level is constant. The raw image can therefore already be regarded as a segmentation, in the worst case each pixel represents a segment as in the region merging techniques described in Chapter 2. Real images however, usually have some regions in which the grey level does not vary in which case an initial segmentation is the partition of flat zones. Throughout the following sections, the segmentation images will be presented in both mean replaced and binary outline form in which pixels on the edges of segments are considered foreground, displayed as black, and those inside the segment background, displayed as white. It should be noted that the outline images are necessarily up-sampled by a factor of three to allow single pixel segments to be outlined accurately, but for presentation they have been rescaled accordingly. In order to enable a direct comparison of the various operators, plots of the coding cost measures are presented as a function of the resulting number of segments and not the operator parameter such as λ or area threshold, a.

Figure 3.7(a) shows a section of the original image of girl which will be used for illustrative purposes throughout this section. It should be noted however that the algorithms have been applied to the whole image (Figure B.1(a)). Figure 4.5 shows the coding cost measures for the image of 'girl' after application of the area open-

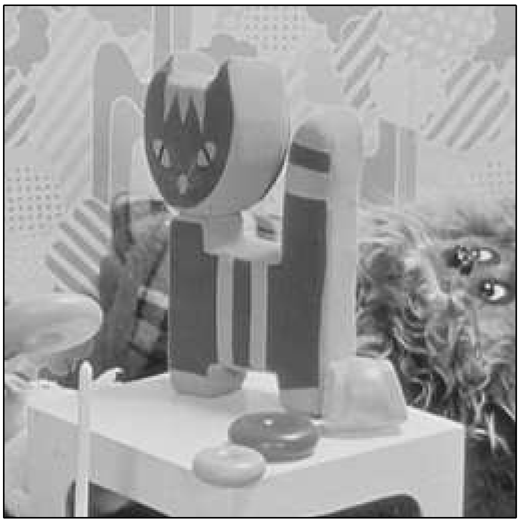


Figure 4.4: A section of the original image of ‘girl’.

closing operator as a function of the number of segments. Since the theoretical upper bound is given by the total number of pixels in the image (414,720 for this 720 × 576 image,) the area thresholds used in the computation of the segmentations are logarithmically spaced at 1, 2, 3, ...10, 20, 30, ...100, 200, 300... etc. pixels. A measurement is included which corresponds to the evaluation of the coding cost measures using the original image as the segmentation image.

The total coding cost curve of Figure 4.5 describes a clear minimum being just below that of the un-segmented image. It has been observed in the application of this technique to other images that the global minimum coding cost occasionally corresponds to that of the un-segmented image. The cost then rises with a decreasing number of segments before dropping again to describe a second, well defined local minimum. It is suggested that more efficient methods of coding the outline image will in some cases reduce this problem, but in any case the initial segmentation can be disregarded as it does not usually represent a useful result. The minimum (circled in Figure 4.5) in this case has a total coding cost of 6.57 bits per pixel and occurs when there are 49,896 segments. This corresponds to an area threshold of 2000 pixels. A section of the mean replaced and outline images for an area threshold of 2000 are given in Figure 4.6.

Figure 4.7 shows the coding cost measures for segmentations produced using the

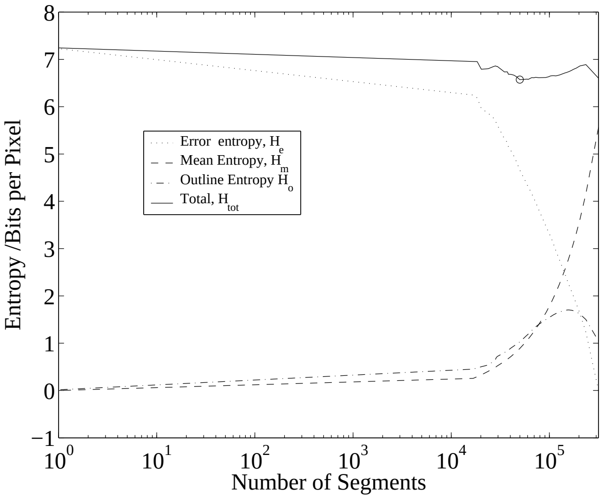


Figure 4.5: The coding cost measures for the area open-closing of the image of 'girl' as a function of the number of segments present in the segmentation.

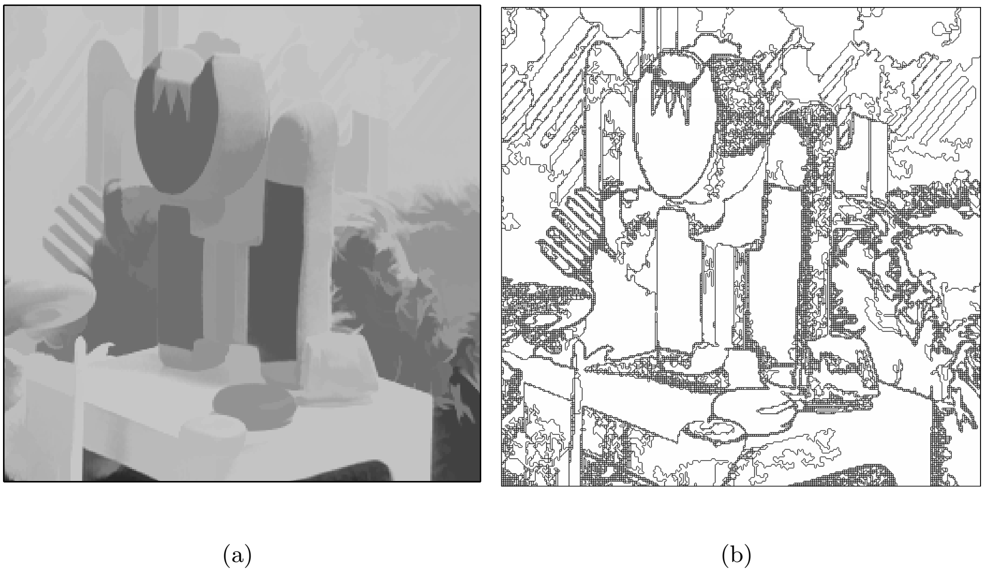


Figure 4.6: The section of the image of 'girl' presented in Figure 4.4 after application of the area open-closing operator with area threshold 2000 (a) and the corresponding outline image (b).

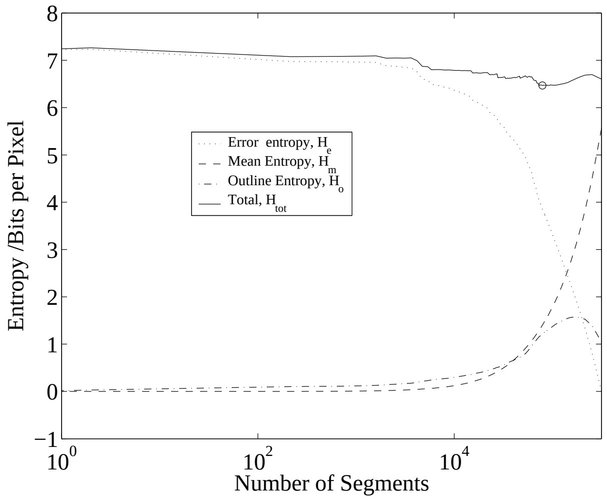
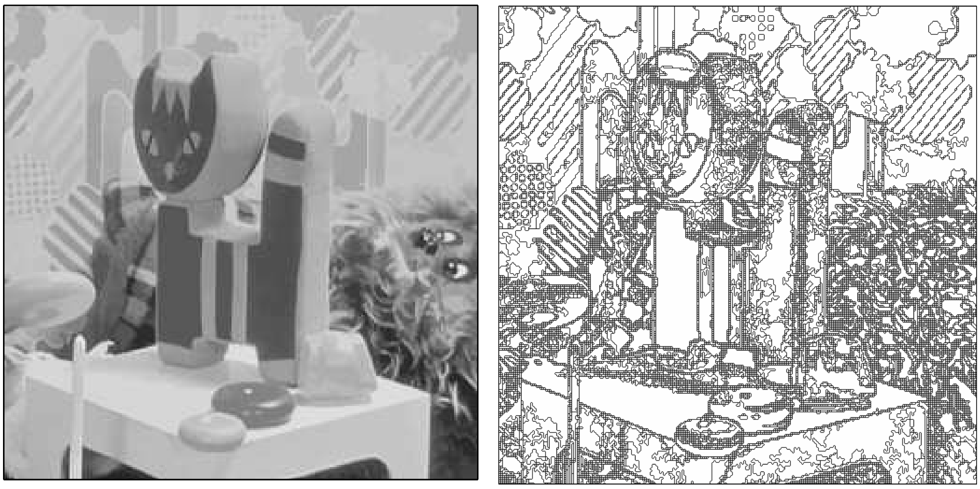


Figure 4.7: The coding cost measures for the λ-max open-closing of the image of 'girl' as a function of λ.

λ-max operator as a function the number of regions. As with the area open-closing, λ is increased from 1 to the value at which only one region remains. Unlike the area open-closing however for an 8-bit image the maximum value for which this is achieved is 128 and as such the chosen values are linearly spaced. In this example, the minimum coding cost of 6.46 bits per pixel is achieved with 79,123 regions using $\lambda = 23$, a section of the mean replaced and outline images for which are given in Figure 4.8.

Figure 4.9 shows a plot of the coding cost measures for the adaptive λ-max operator. The minimum in the total coding cost has a value of 6.39 bits per pixel with 90,115 regions for which the corresponding value of λ is 11. A section of the mean replaced and outline images are presented in Figure 4.10. A comparison of the total coding cost of all three connected operators is presented in Figure 4.11 from which we see clearly that the lowest coding cost is achieved by the adaptive λ-max operator but this minimum is associated with a larger number of regions than the minima of the other methods. In defining the desirable properties of a segmentation metric we required it to allow a direct, quantitative comparison of segmentation algorithms as well as the optimisation of algorithm parameters. We have also stated that the properties of a segmentation, and in particular the number of segments, will vary depending on the application. Therefore, in order to compare algorithms it is only



(a) (b)

Figure 4.8: The section of the image of 'girl' presented in Figure 4.4 after application of the λ-max open-closing operator with $\lambda = 23$ (a) and the corresponding outline image (b).

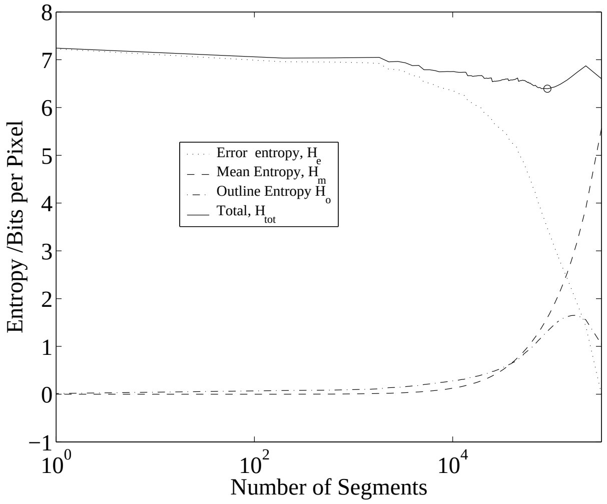


Figure 4.9: The coding cost measures for the adaptive λ-max open-closing of the image of 'girl' as a function of λ.

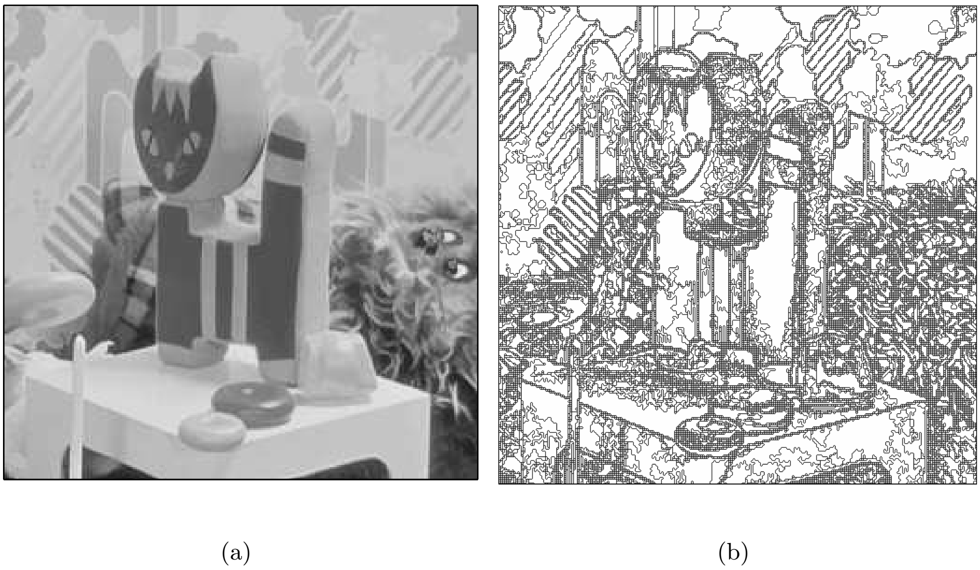

(a) (b)

Figure 4.10: The section of the image of ‘girl’ presented in Figure 4.4 after application of the adaptive λ-max open-closing operator with $\lambda = 11$ (a) and the corresponding outline image (b).

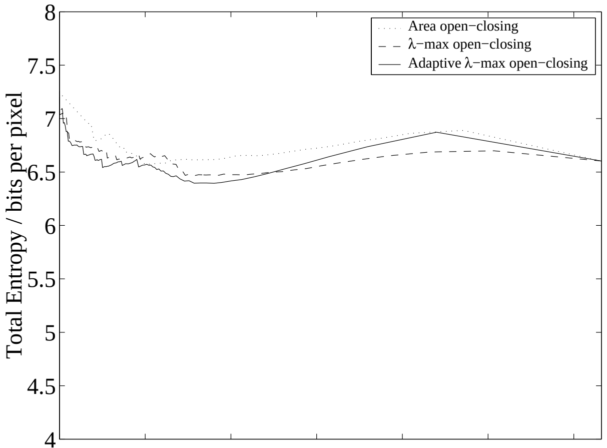


Figure 4.11: The total coding cost measures for the area, λ-max and adaptive λ-max open-closing of the image of ‘girl’ as a function of the number of segments.

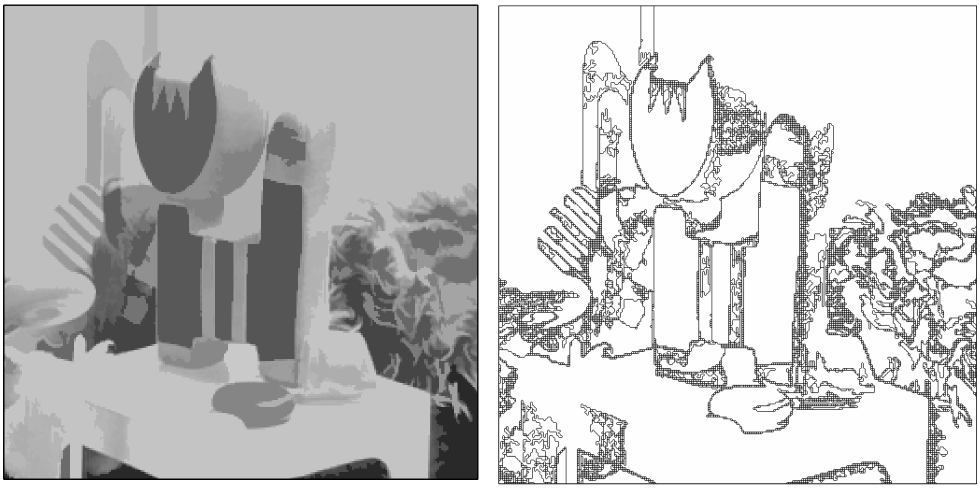
reasonable to compare the total coding costs of segmentations with equal numbers of segments. Figure 4.11 shows that for segmentations with fewer than 125,000 segments the adaptive λ-max operator yields lower coding costs while for $N_s > 125,000$ the λ-max operator performs better. Since in the majority of applications, a segmentation with 125,000 segments will be considered a vast over-segmentation we can conclude that the adaptive λ-max operator is most suitable for the segmentation of this image.

We note from the outline images presented in this section that the suppression of regional extrema leads to two distinct types of region in the segmentation; flat zones, described by the larger segments, and transition zones in which large numbers, of often single pixel segments, proliferate. The flat zones correspond well with the interiors of objects, but the transition zones prevent the boundaries of these segments from corresponding well with object boundaries. The existence of these transition zones is due to the corresponding pixels of the original image lying in regions which are neither maxima or minima such as low gradient edges at object boundaries, and as such are unaffected by the connected operator until the parameter (area threshold or λ) is increased to such an extent that they are included in the larger flat zones. Figure 4.12 shows the mean replaced and outline images for the segmentation achieved by applying the adaptive λ-max operator with $\lambda = 50$. We note that whilst the transition regions have been reduced, the flat regions have increased so as to breach significant object boundaries such as the top edge of the toy on the table.

In the following section we propose a number of new connected operators designed specifically to clean up the transition zones resulting in a reduction in the number of regions and better correspondence of segments boundaries with the edges of significant regions. It will also be shown that these post-processing connected operators improve the performance of the segmentations with respect to the metric.

4.3 New Connected Operators for Post-processing

In the previous section it was demonstrated that the process of suppressing regional extrema has the effect of producing flat regions which correspond well with the interiors of significant regions of the image scene and transition zones which tend to



(a) (b)

Figure 4.12: The section of the image of ‘girl’ presented in Figure 4.4 after application of the adaptive λ-max open-closing operator with $\lambda = 50$ (a) and the corresponding outline image (b).

form large ‘belts’ of small segments at object boundaries. It was demonstrated that there is a trade off between the size of the transition zones and the tendency for the regions to leak between significant areas. In the following section we propose two connected operators which aim to clean up the segmentation by merging regions smaller than a given threshold with their neighbours regardless of their topology.

The first operator we propose scans the image left to right top to bottom merging any flat region consisting of a single pixel with its largest neighbouring flat region. If a flat region has no neighbours larger than itself, it is likely to be in the middle of a transition zone and is left for the next iteration. The process is continued until idempotence, i.e. all single pixel regions have been merged with their largest neighbours. The size of the flat regions to merge is then increased by one pixel and the whole process repeated. This continues until a specified area threshold a has been met. The algorithm is written in pseudo-code below.

```
for size=1,2,...a
{
  repeat to idempotence
  {
```

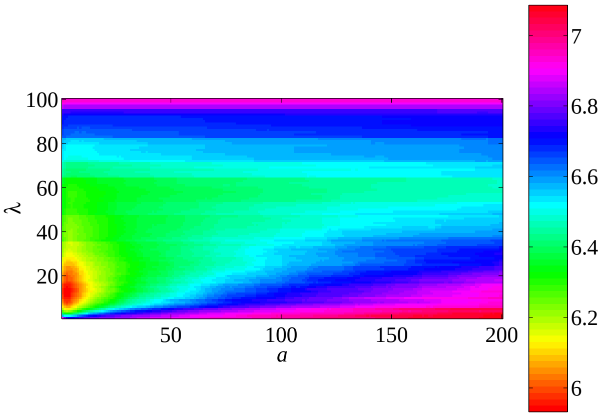
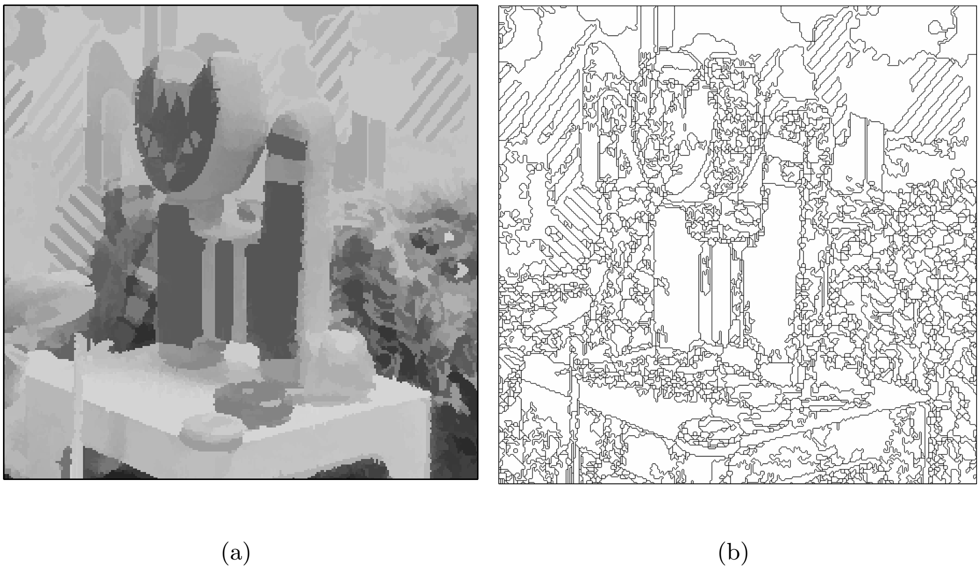


Figure 4.13: The total coding cost as a function of λ and a for the segmentation of the image 'girl' processed using the adaptive λ-max and area-merging algorithms.

```
    for each flat region in the image
    {
      if current region size < size & ...
         current region has at least one neighbour of size > size
      {
        -merge current region with largest neighbour
      }
    }
  }
}
```

The effect of the algorithm is to gradually erode the transition regions iteratively merging them with the larger flat regions which as we have seen correspond well with objects in the scene. It is hoped that by merging only a single layer of small segments on each iteration the flat zones will eventually meet at the boundaries of significant regions in the image.

Figure 4.13 shows the result of applying the area-merging connected operator to the results of the adaptive λ-max operator, chosen due to the low coding cost achieved by the operator in the previous section. There are now two parameters to the segmentation, λ and a, and as such the results are presented as an image with the colour denoting the total coding cost value. The minimum total coding cost for the image of 'girl' is 5.84 bits per pixels and occurs at $\lambda = 12$, $a = 8$ with 6,625 segments. Though an exhaustive computation of the total coding cost of segmentations produced by different values of λ and a is presented here, the location of the minimum suggests that a suitable segmentation may be identified by alternate projection op-



(a) (b)

Figure 4.14: The section of the image of 'girl' presented in Figure 4.4 after application of the adaptive λ-max open-closing operator with $\lambda = 14$ and the area-merging connected operator with $a = 3$ (a) and the corresponding outline image (b).

timisation, choosing first a suitable value for λ by application of the adaptive λ-max operator only and then applying the area-merging operator to determine an optimal value of a. It is clear that by introducing the area-merging post-processing step we have significantly reduced both the total coding cost and the number of of segments associated with the optimal segmentation.

Figure 4.14 shows the mean replaced and outline image of the segmentation which corresponds to the minimum of Figure 4.13. We note from these images that the edges of segments are not as tidy as we may wish them to be, suggesting that some segments in the transition regions have been merged with the wrong flat region. The second connected operator proposed here attempts to overcome this by merging segments in the transition zones with flat segments based upon their relative grey levels.

The pseudo-code for the second operator is identical to the first, but rather than merging a flat region of less than $1, 2, ...a$ pixels with the largest neighbouring flat region, the small region is merged with the neighbouring segment with the closest grey level. Note that here the grey levels of segments are defined by the connected

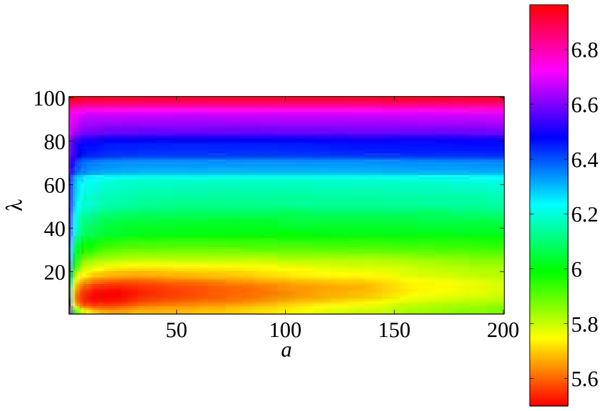
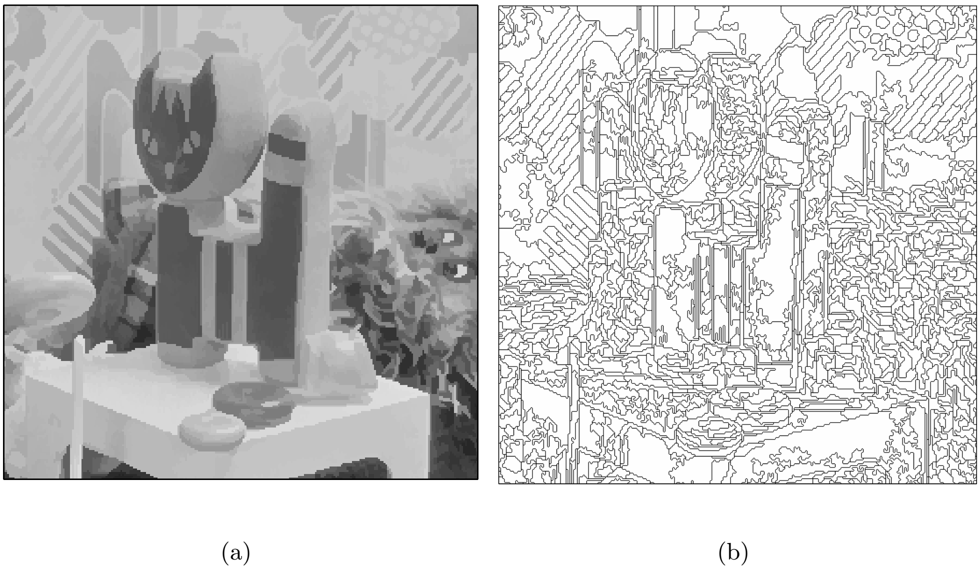


Figure 4.15: The total coding cost as a function of λ and a for the segmentation of the image 'girl' processed using the adaptive λ-max and grey scale-merging algorithms.

operator and not the mean value. It is hoped that in so doing, large segment boundaries will converge on the grey level discontinuities and therefore object boundaries which lie somewhere inside the transition zones more accurately than the previous operator. Again the process is iterated for each area threshold until idempotence. Figure 4.15 shows the total coding cost as a function of both λ and a for the segmentation of the image 'girl' processed using the adaptive λ-max and grey scale-merging algorithms. Note that the colour scaling of Figures 4.13 and 4.15 are not the same.



(a) (b)

Figure 4.16: The mean replaced (a) and outline image (b) of a section of the segmentation of the image of ‘girl’ after application of the adaptive λ-max open-closing operator with $\lambda = 8$ and the area-merging connected operator with $a = 20$ (a).

The minimum total coding cost using the adaptive λ-max and grey scale-merging algorithms is 5.50 bits per pixel and occurs when $\lambda = 8$, $a = 20$ and the resulting segmentation has 5,651 segments. Such a low coding cost suggests that using this scheme the image could potentially be compressed without loss by values approaching 31% its original size. This compares favourably with propriatory lossless file compression utilities which, for the image ‘girl’ in 8-bit RAW[1] format can achieve 23% compression. The mean replaced and outline images corresponding to this minimum are given in Figure 4.16. The minimum of the surface occurs at such a low value of λ as to suggest that the application of the adaptive λ-max operator is unnecessary, however, Figure 4.17 shows the coding cost measures after applying the grey level-merging operator only on the image ‘girl’ the minimum of which has a total coding cost of 5.69 bits per pixel.

Whilst many of the segment edges of Figure 4.16 correspond well with object edges, the segmentation corresponding to the minimum total coding cost may still be re-

[1] The RAW format as its name suggests consists of the pixel values written in raster scan, no header information, such as image dimentions is saved.

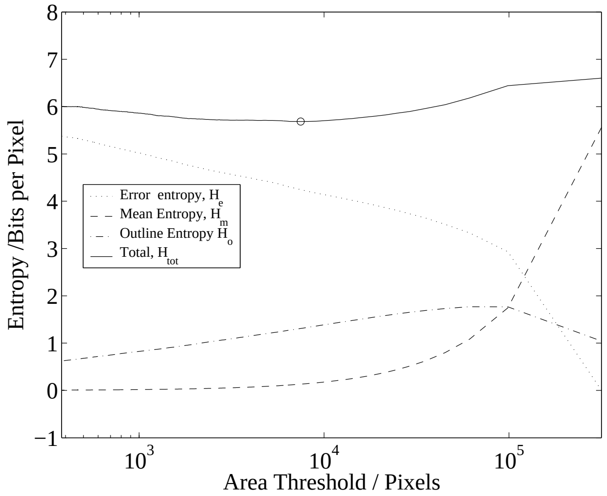


Figure 4.17: The coding cost measures for the grey scale-merging connected operator only as a function of area threshold a.

garded as an over-segmentation. As such we propose the introduction of a constant, α into the calculation of the total coding cost which penalises segmentations with large numbers of segments by increasing the relative importance of both the segment means coding cost C_m and the outline coding cost C_o. The weighted total coding cost is now calculated according to the following.

$$C_{tot,w} = \alpha(C_m + C_o) + (1 - \alpha)C_e \tag{4.10}$$

A value of $\alpha = 0.5$ will result in values for λ and a as seen previously, however increasing the value of α will tend to shift the minimum of the total coding cost towards a result with a smaller number of segments. This method allows the selection of a suitable segmentation dependent on only one parameter which can be tuned for a particular image and / or application. Figure 4.18 shows the weighted total coding cost as a function of λ and a for the segmentation of the image 'girl' using $\alpha = 0.6$, from which it is clear that the minima is located at higher values of both λ and a when compared with Figure 4.15. It should be noted that the colour information in Figure 4.18 has been clipped to allow resolution of the minimum.

Figures 4.19 and 4.20 show sections of the mean replaced and outline images for the segmentations of the image of 'girl' which correspond to the minima of the weighted total coding cost surface for $\alpha = 0.6$ and $\alpha = 0.75$ respectively. Values of the number of segments, un-weighted coding costs, λ and a are presented along with

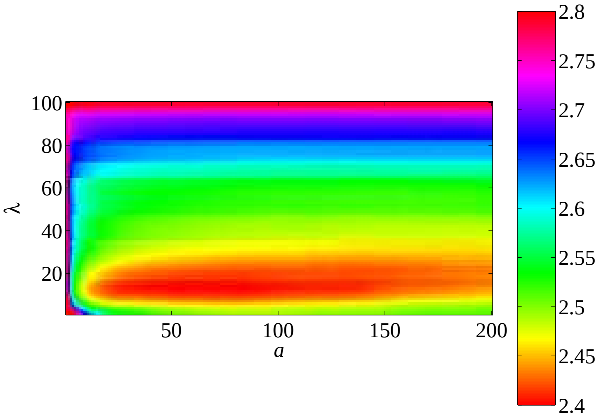


Figure 4.18: The weighted total coding cost as a function of λ and a for the segmentation of the image 'girl' processed using the adaptive λ-max and grey scale-merging algorithms with $\alpha = 0.6$.

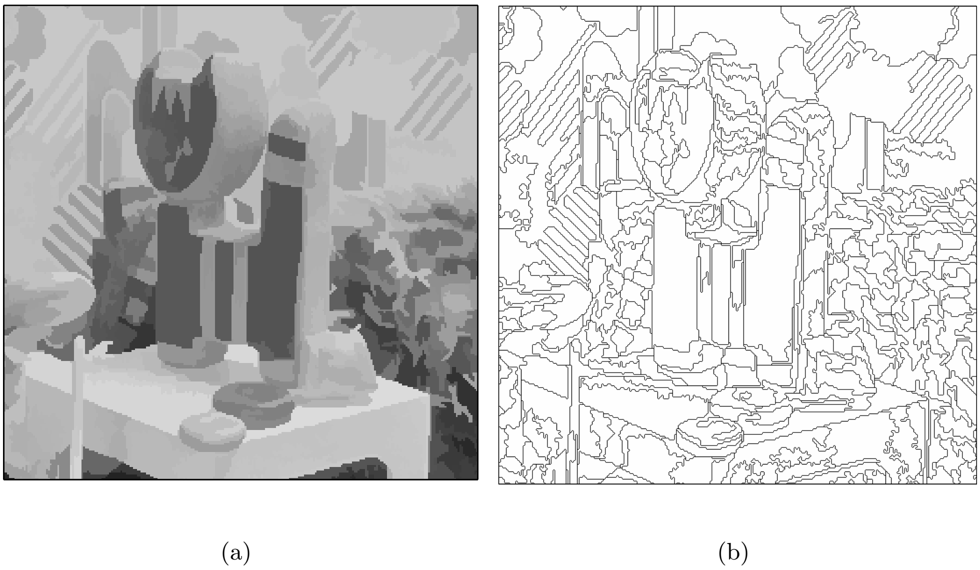
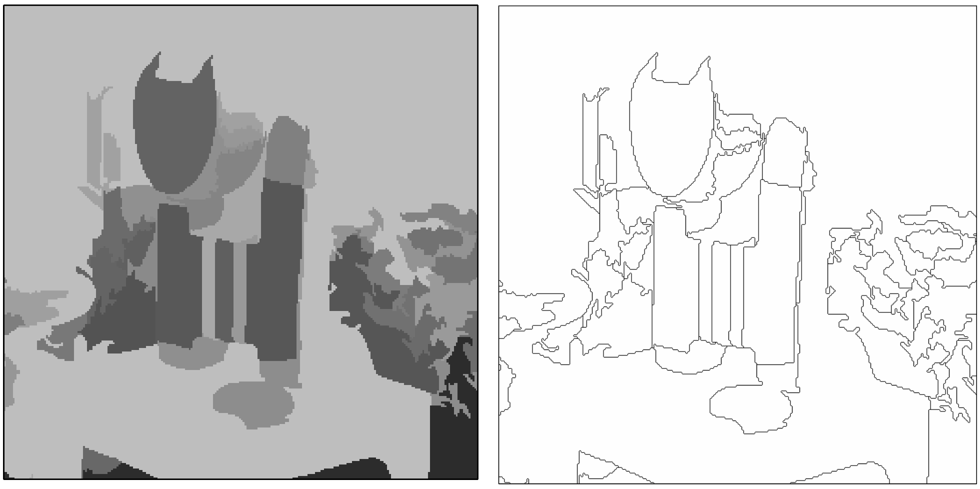


Figure 4.19: The mean replaced (a) and outline image (b) of a section of the segmentation of the image of 'girl' produced using the adaptive λ-max and grey scale-merging operators. The result presented corresponds to the minimum of the weighted total coding cost surface for $\alpha = 0.6$.



(a) (b)

Figure 4.20: The mean replaced (a) and outline image (b) of a section of the segmentation of the image of 'girl' produced using the adaptive λ-max and grey scale-merging operators. The result presented corresponds to the minimum of the weighted total coding cost surface for $\alpha = 0.75$.

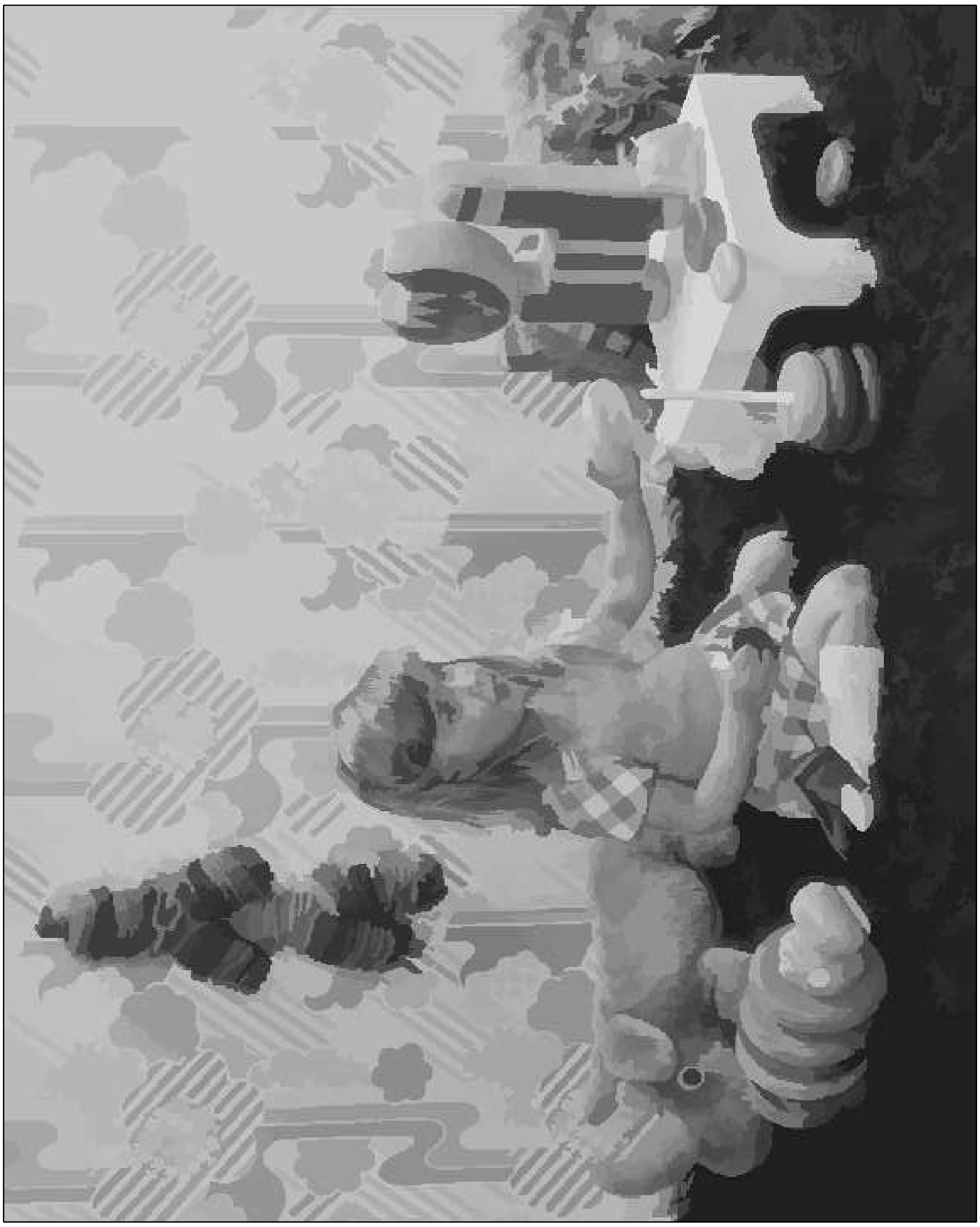
similar measures for other images in Table 4.1.

Figure 4.21 shows mean replaced and outline image of the complete segmentation corresponding to the minimum weighted coding cost for the adaptive λ-max and grey scale-merging operators applied to the image of 'girl' with $\alpha = 0.6$. The resulting values of λ and a are 14 and 80 respectively. Results for other images are presented in Appendix C. Note that no attempt has been made to 'tune' α to give subjectively pleasing results but that the segmentation results for a given value of α are highly dependant upon the image content. A value of $\alpha = 0.75$ for the image of 'boat' for example, results in a single region (and is therefore not presented in the Appendix) whilst the same value for the image of 'lenna' results in 55 segments. We note from Table 4.1 that increasing the value of α above 0.5 has the effect of decreasing the number of segments in the optimal segmentation of all our test images coupled with a corresponding increase in the un-weighted coding costs. We also observe increases in both λ and a. With regard to the objectives of this thesis, the segmentations resulting from the adaptive λ-max and grey level-merging operators (see Appendix C and Figures 4.16, 4.19, 4.20 and 4.21) comprise connected regions of pixels whose boundaries (with the correct tuning of α) correspond well with object boundaries. We can see that in cases where the contrast is low, such as the image of 'toys', a lower value of α may be required in order to prevent segments from breaching object boundaries.

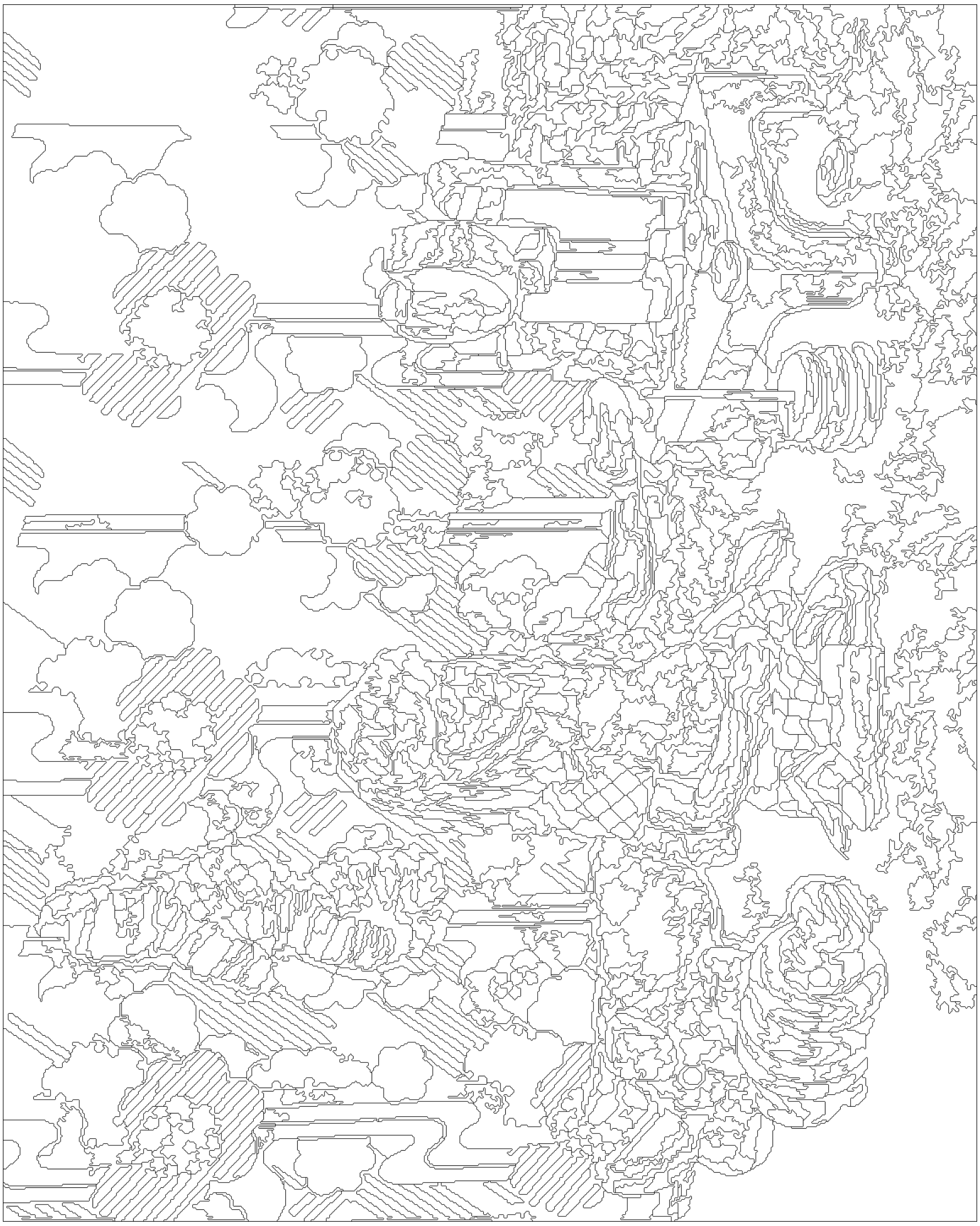
In the following section we introduce an operator which aims to minimise our evaluation metric by using a drop in the segmentations coding cost as a criteria for merging regions which are initialised as the partition of flat zones.

4.4 A Coding Cost Minimising Connected Operator

Starting with the partition of flat zones as the initial segmentation the algorithm associates with each segment a value equal to the mean of the image inside that segment. Scanning the segments from left to right, top to bottom, each segment is merged with each of its direct neighbours individually and the rounded mean of the new region evaluated. It may be the case that a number of regions neighbouring the new one have the same mean as the newly calculated one and as such these are also



(a)



(b)

Figure 4.21: The image of 'girl' after application of the adaptive λ-max open-closing and the grey scale-merging connected operator with $\alpha = 0.6$ (a) and the corresponding outline image (b).

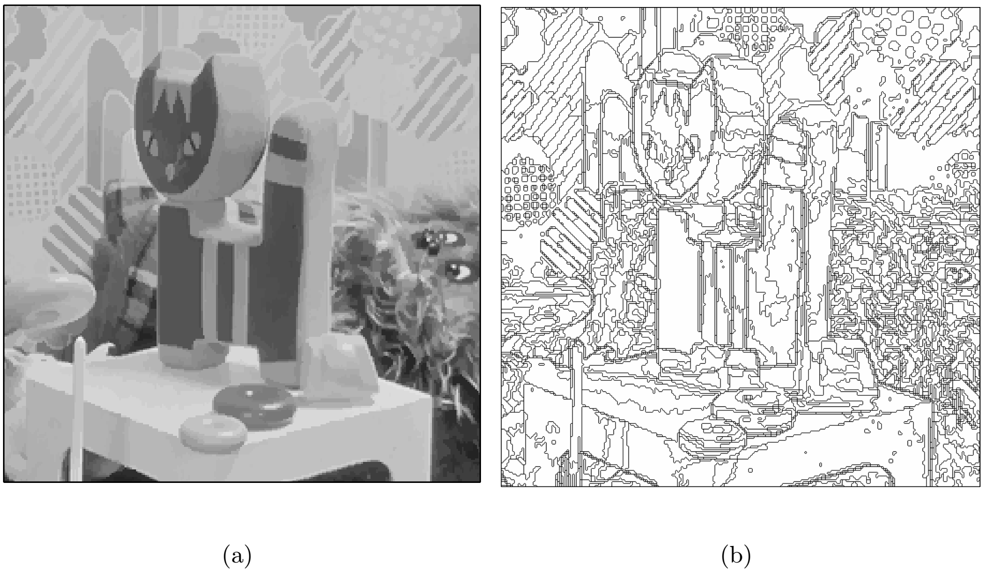
	No. of Segments	Coding Cost (C_{tot})	λ / a
'boat' (256 × 256)			
Original	50,501	6.384	- / -
$\alpha = 0.5$	998	6.087	13 / 4
$\alpha = 0.6$	216	6.132	30 / 44
$\alpha = 0.75$	1	6.764	96 / 1
'clown' (256 × 256)			
Original	45,792	7.119	- / -
$\alpha = 0.5$	2,447	5.529	6 / 8
$\alpha = 0.6$	1,484	5.643	9 / 13
$\alpha = 0.75$	30	6.531	61 /185
'girl' (720 × 576)			
Original	316,108	7.227	- / -
$\alpha = 0.5$	5,651	5.500	8 / 20
$\alpha = 0.6$	1,306	5.614	14 / 80
$\alpha = 0.75$	164	6.181	58 /199
'lenna' (256 × 256)			
Original	54,398	7.572	- / -
$\alpha = 0.5$	725	6.191	16 / 30
$\alpha = 0.6$	253	6.235	23 / 83
$\alpha = 0.75$	55	6.563	60 /197
'toys' (256 × 256)			
Original	53,449	6.943	- / -
$\alpha = 0.5$	725	5.739	12 / 31
$\alpha = 0.6$	329	5.799	19 / 63
$\alpha = 0.75$	9	6.601	80 /192
'tree' (256 × 256)			
Original	62,976	7.616	- / -
$\alpha = 0.5$	54	7.510	74 /144
$\alpha = 0.6$	28	7.550	87 /198
$\alpha = 0.75$	15	7.664	100/195

Table 4.1: A table of the number of segments, un-weighted coding cost and values of λ and a for the adaptive λ-max and grey scale-merging connected operators evaluated for values of $\alpha = 0.5,\ 0.6$ and 0.75 acting on various images. The resulting segmentations are presented in Appendix C.

included in the new region. The total coding cost is then evaluated for each merge and the merge giving the greatest reduction is kept for the next iteration. The new region is marked as 'merged' to prevent subsequent regions merging with it. If none of the merges yields a reduction in the total coding cost then no merge is made. In this way, each region is permitted to merge with at most one neighbour (not including those with the same mean value) on each iteration. Once all regions in the segmentation have been scanned and possibly merged, the process begins again. This continues to idempotence, i.e. no more regions can be merged to yield a drop in the total coding cost. Note that this method is not optimal as to be so would require an exhaustive test of all possible segment merges which is not practicable for even modest numbers of segments. The pseudo-code for the algorithm is given below.

```
Beginning with the partition of flat zones (current segmentation):
do
{
-repeat flag=0
  for each segment (current segment)
  {
    -add to stack all neighbouring segments
    -current best segmentation=current segmentation
    while stack is not empty
    {
     -merge current segment with segment on stack
     -merge new segment neighbours with identical means
           (new segmentation)
     -evaluate total coding cost of new segmentation
     if total coding cost of new segmentation <=
           total coding cost of current best segmentation
     {
       -current best segmentation=new segmentation
       -repeat flag=1
     }
    }
    -current segmentation= current best segmentation
  }
}
while repeat flag==1
```

Figure 4.22 shows sections of the mean replaced image and segment outline resulting from the coding cost minimising operator for the image of 'girl'. The segmentation has 14,000 segments and a total coding cost of 5.294 bits per pixel. This compares



(a) (b)

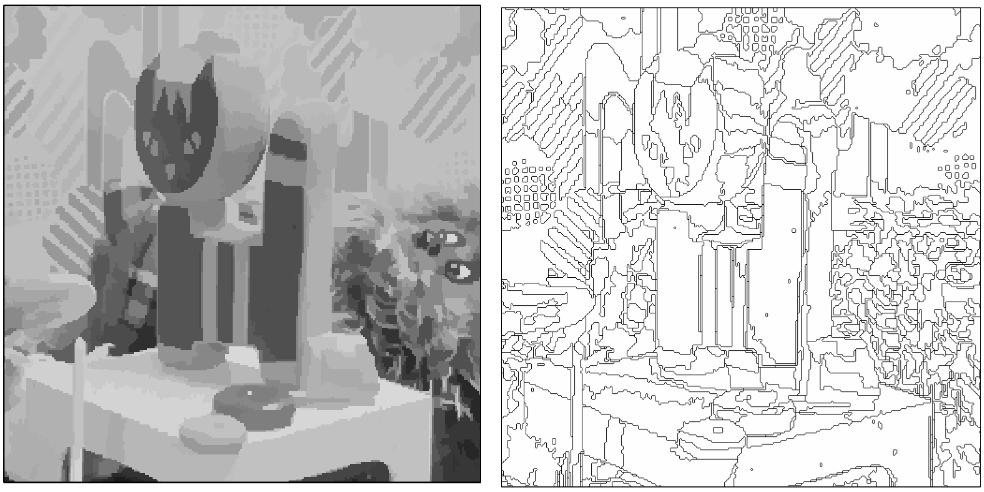
Figure 4.22: Sections of the mean replaced (a) and segment outline (b) of the image of ‘girl’ after application of the coding cost minimising operator.

favourably with the un-weighted coding cost of the adaptive λ-max and grey scale-merging scheme of the previous chapter, but has significantly more regions.

The number of segments can be reduced by evaluating the weighted total coding cost according to equation 4.10 with $\alpha > 0.5$, thus penalising large numbers of segments and increasing the tendency of the algorithm to merge segments. Sections of the mean replaced and outline images of the segmentations due to the operator acting on the image of ‘girl’ for $\alpha = 0.6$ and $\alpha = 0.75$ are presented in Figures 4.23 and 4.24 respectively from which we see that increasing α has had the effect of reducing the number of segments whilst maintaining their correspondence with object boundaries. The coding costs and exact number of segments for these segmentations are presented in Table 4.2.

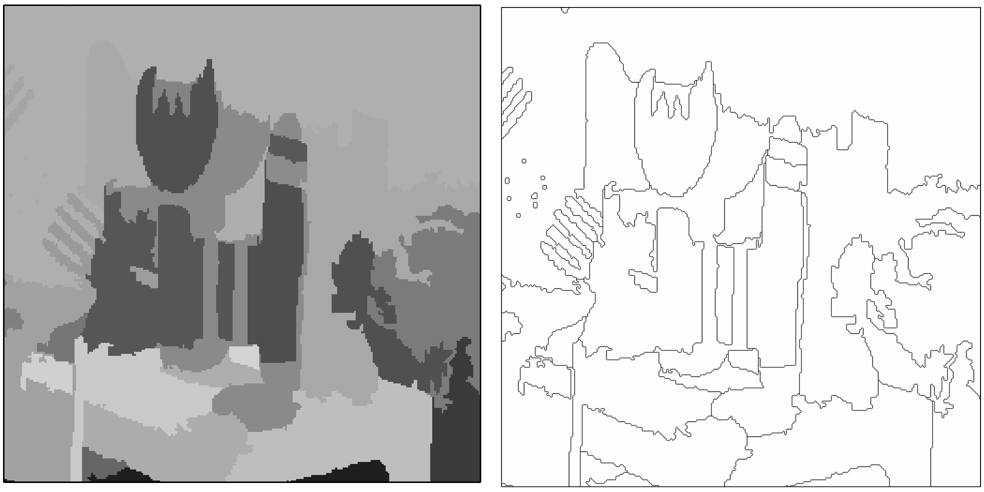
The complete mean replace and outline images of the segmentation resulting from the application of the weighted coding cost minimising operator on the image of ‘girl’ with $\alpha = 0.6$ are presented in Figure 4.25.

More examples of the coding cost minimising connected operator are given in Appendix D. The results correspond to values of $\alpha = 0.5, 0.6$ and 0.75 which have been



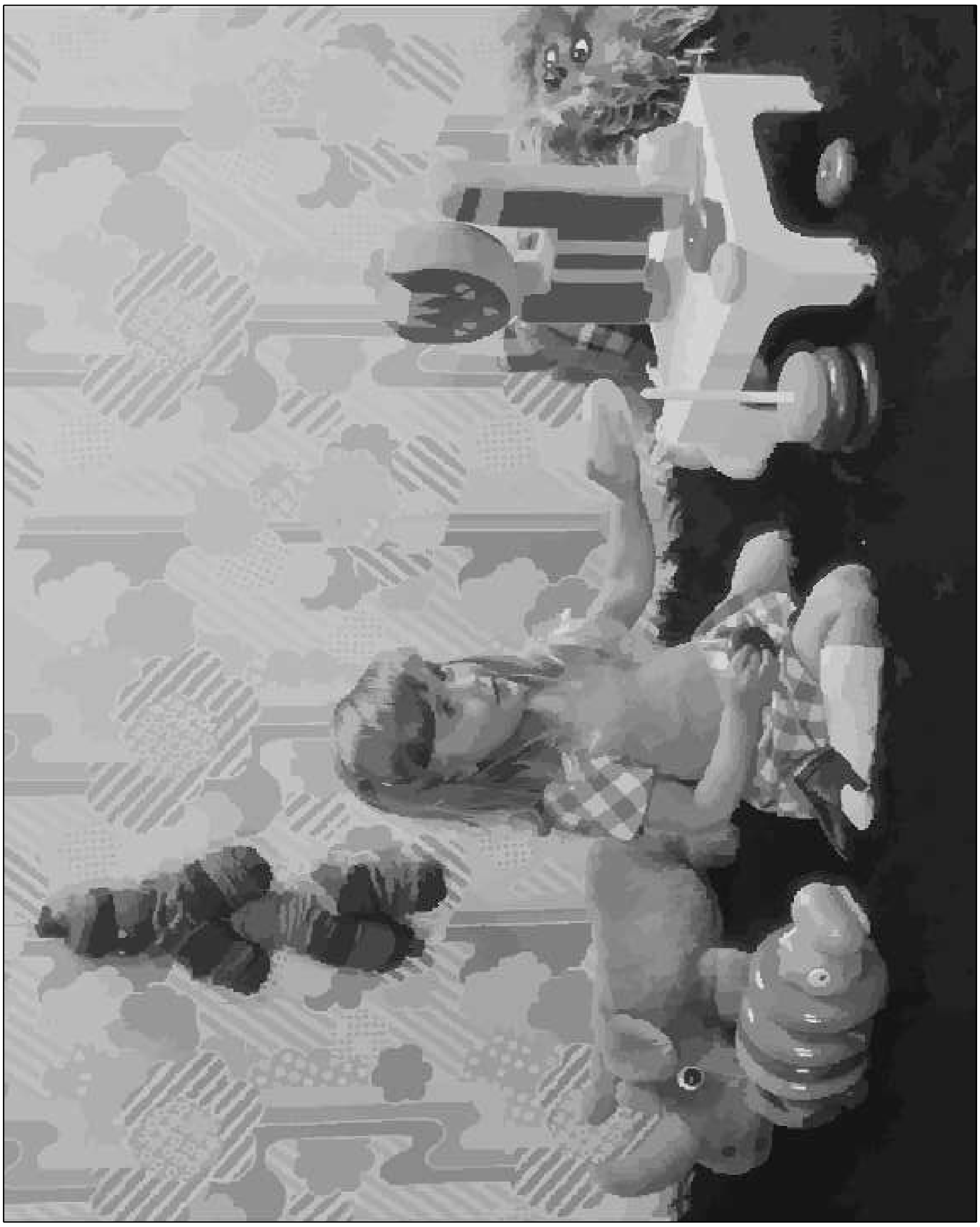
(a) (b)

Figure 4.23: A section of the mean replaced (a) and outline images (b) due to the segmentation of the image ‘girl’ using the weighted coding cost minimising operator with $\alpha = 0.6$.

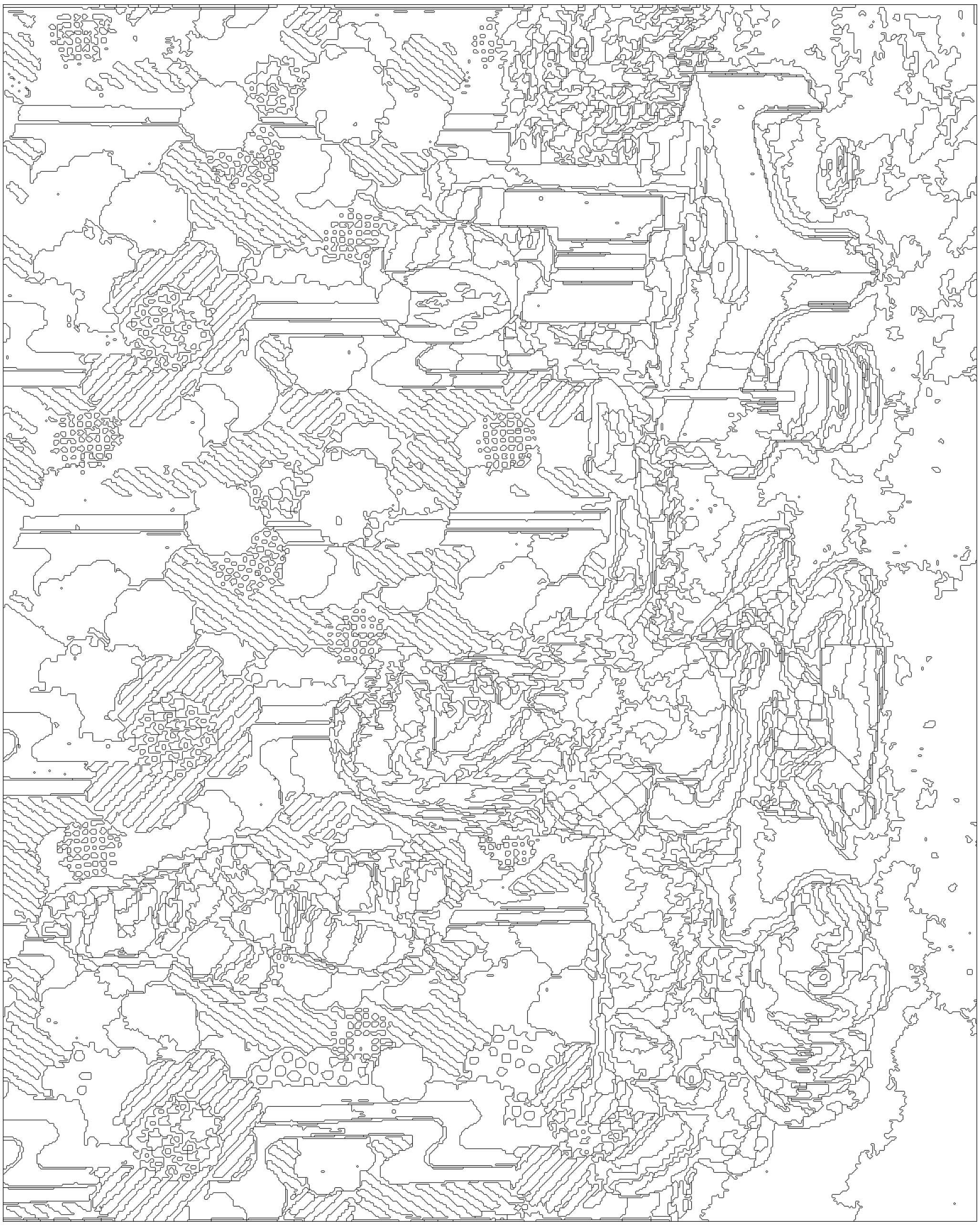


(a) (b)

Figure 4.24: A section of the mean replaced (a) and outline images (b) due to the segmentation of the image ‘girl’ using the weighted coding cost minimising operator with $\alpha = 0.75$.



(a)



(b)

Figure 4.25: The mean replaced image of ‘girl’ after application of the weighted coding cost minimising operator with $\alpha = 0.6$ (a) and the corresponding outline image (b).

	No. of Segments	Coding Cost (C_{tot})	No. of Iterations
‘boat’ (256 × 256)			
Original	50,501	6.384	-
$\alpha = 0.5$	30,358	6.126	40
$\alpha = 0.6$	631	5.881	50
$\alpha = 0.75$	46	6.325	321
‘clown’ (256 × 256)			
Original	45,792	7.119	-
$\alpha = 0.5$	7,914	5.298	40
$\alpha = 0.6$	1,217	5.359	72
$\alpha = 0.75$	97	6.197	254
‘girl’ (720 × 576)			
Original	316,108	7.227	-
$\alpha = 0.5$	14,000	5.294	70
$\alpha = 0.6$	2,807	5.410	125
$\alpha = 0.75$	140	6.130	909
‘lenna’ (256 × 256)			
Original	54,398	7.572	-
$\alpha = 0.5$	4,151	5.914	53
$\alpha = 0.6$	502	5.992	72
$\alpha = 0.75$	38	6.572	74
‘toys’ (256 × 256)			
Original	53,449	6.943	-
$\alpha = 0.5$	1,890	5.508	40
$\alpha = 0.6$	320	5.618	36
$\alpha = 0.75$	47	6.153	83
‘tree’ (256 × 256)			
Original	62,976	7.616	-
$\alpha = 0.5$	62,976	7.616	-
$\alpha = 0.6$	62,552	7,559	15
$\alpha = 0.75$	1	7.649	310

Table 4.2: A table of the number of segments, un-weighted coding cost and number of iterations required for the coding cost minimising connected operator evaluated for values of $\alpha = 0.5,\ 0.6$ and 0.75 acting on various images. The resulting segmentations are presented in Appendix D.

chosen arbitrarily, though of course the degree of segmentation required is dependant on the image content. It is interesting to observe that perceptually the mean replaced images for $\alpha = 0.5$ do not differ significantly from the originals suggesting that the error image may be replaced by dither or even discarded completely if a lower coding cost were sought for image compression applications. Based upon these results Table 4.2 presents values for the number of segments and the un-weighted total coding cost C_{tot} and the number of iterations to idempotence for the three segmentations of different images, the values for the original image are also included. Note that for the image of 'boat' the total coding cost for $\alpha = 0.6$ is lower than that for $\alpha = 0.5$, highlighting the sub-optimal nature of the algorithm. Also, mean replaced and outline images of the segmentations of 'tree' for $\alpha = 0.75$ have been omitted as the resulting segmentation consists of a single segment suggesting that a suitable value of α for this highly textured image may lie in the range $0.6 < \alpha < 0.75$. Comparison of Tables 4.1 and 4.2 reveals that in most cases the un-weighted coding cost of the segmentations produced using the coding cost minimising operator are lower than those of the adaptive λ-max open-closing and grey scale-merging operators as one might expect. This said, the results presented in Appendix D and Figures 4.22, 4.23 and 4.24 show that though the segments are connected and their boundaries correspond relatively well with significant regions, the correspondence is not as high as that achieved by the results of the adaptive λ-max and grey scale-merging operators. This could potentially be improved by imposing further constraints on the merging of regions such as a grey level distance threshold as in the grey level-merging algorithm but we leave this for future work. Also recommended for further investigation are methods for improving the computational efficiency as in its current form the algorithm is considerably slower than the other techniques presented.

4.5 Summary

This chapter has introduced a novel metric for the evaluation of an arbitrary partial segmentation based upon the cost of coding the image using that segmentation. It was then shown, using this metric to identify the 'best' segmentation that the connected operators of area, λ-max and adaptive λ-max open-closing are applicable to image segmentation but the existence of low gradient edges results in highly over-segmented regions which we have referred to as transition zones. A pair of novel post-processing operators which fit into the definition of connected operators were then proposed for the mitigation of said transition zones. Of these, the grey scale-merging operator was shown to perform well in terms of the reduction in coding cost and the location of segment boundaries. The metric was modified to allow for a tuning parameter α which indirectly controls the number of segments in the segmentation corresponding to the minimum of the weighted total coding cost surface. This was shown to improve the performance of connected operators according to the criteria laid down in Chapter 2. A novel connected operator in the form the coding cost minimising operator was developed using the total coding cost of the segmentation as a merging criterion and it was shown that the results were encouraging. Finally the operator was modified using the weighted evaluation metric and shown to give results which outperform those obtained previously in terms of the un-weighted total coding cost but in which the segment boundaries correspond less well with the edged of homogeneous regions.

In the sequel, we introduce two novel schemes for image segmentation based on improving the watershed transform presented in Section 2.4 using connected operators.

Chapter 5

Improving the Watershed Transform using Connected Operators

In Chapter 2 we introduced the watershed algorithm and demonstrated its tendency to over-segment the image, particularly in the presence of noise. In this chapter we propose two distinct methods which, using connected operators reduce this over-segmentation. The first involves using the now familiar adaptive λ-max operator as a pre-processing stage, reducing the level of noise and therefore the number of non-zero values in the gradient image. We compare this technique with those of linear Gaussian and median filtering. The second uses the operator to smooth the gradient image directly and again the results are compared with traditional image smoothing filters.

5.1 Image Smoothing

The watershed transform associates with every regional minima a catchment basin and in so doing defines a region of the resulting segmentation. It is not surprising then, that in noisy images with many small changes in grey level, and hence regional minima in the gradient image, that the transform has a tendency to over-segment. By reducing the noise in the image prior to calculation of the gradient, we can reduce this tendency. We propose the use of the adaptive λ-max connected operator for this purpose and compare the results, in the context of our metric and objectives, with those produced by both Gaussian linear and median filters. We choose the adaptive λ-max open-closing over the area and λ-max open-closings because it converges to flat zones and as such removes image detail more quickly in terms of increasing λ.

The details of the three strategies for segmentation we shall consider are as follows. Gaussian linear filtering is performed by convolving the image with a two dimensional, square Gaussian kernel of size $n_l \times n_l$, $0 \leq n_l \leq 100$. The kernel is computed such that the Gaussian has decayed by approximately 40dB before truncation. In calculating the median filtered image, the median grey level is taken over a square neighbourhood of size $n_m \times n_m$, $0 \leq n_m \leq 100$. Values of λ in the adaptive λ-max open-closing are taken to be such that $0 \leq \lambda \leq 100$. It should be noted that the maximum size of n_l, n_m and λ do not necessarily produce results with a single segment but experimentation has shown that further increase produces no significant improvement in the results. After smoothing the image the morphological gradient, being the minimum value in a 3×3 neighbourhood subtracted from the maximum value in the same neighbourhood, is calculated. This method follows that described in [17] and Section 2.4. The watershed transform is then performed using the algorithm presented in [20] a discussion of which is also given in 2.4.

Figure 5.1 shows the total coding cost as calculated according to Equation 4.9 as a function of the number of regions for the segmentations of the image of 'girl'. Note that the plot is presented in the form of a scatter chart owing to the fact that linear and median filters of different sizes are capable of producing segmentations with identical numbers of regions. We note from this diagram that the adaptive λ-max open-closing produces the lowest total coding cost of the three schemes. This distinction is confirmed by observation of the mean replaced and outline images of the segmentations for the three schemes corresponding to the respective minima, Figures 5.2, 5.3 and 5.4.

Many of the segment boundaries in all these images correspond well with object boundaries, however for some applications all the results may be regarded as over-

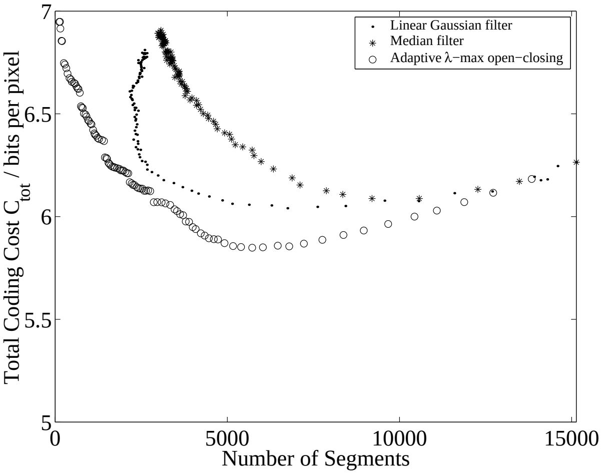


Figure 5.1: The total coding cost C_{tot} of the segmentations of the image 'girl' produced by smoothing the image using linear Gaussian, median and adaptive λ-max open-closing prior to computation of the gradient and watersheds, plotted as a function of the number of segments.

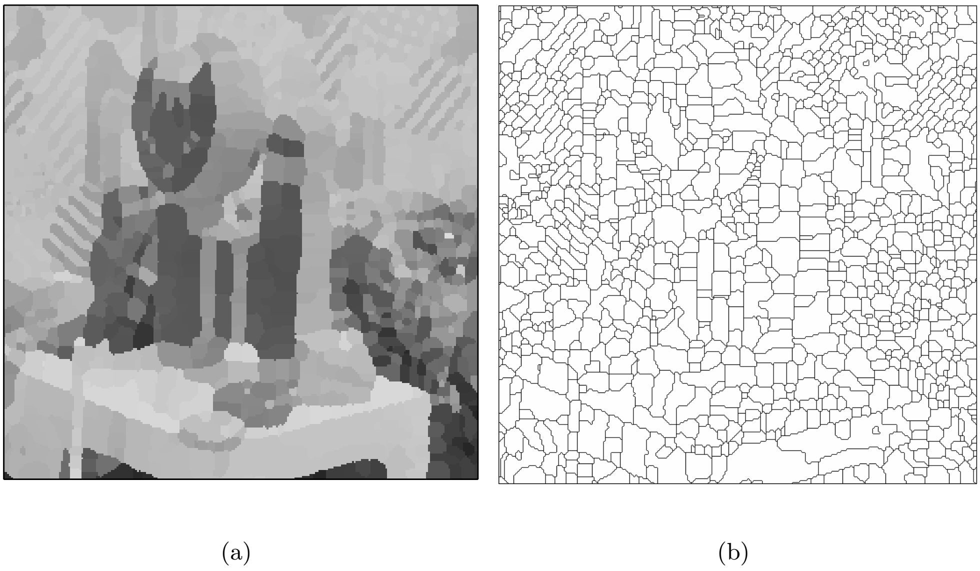
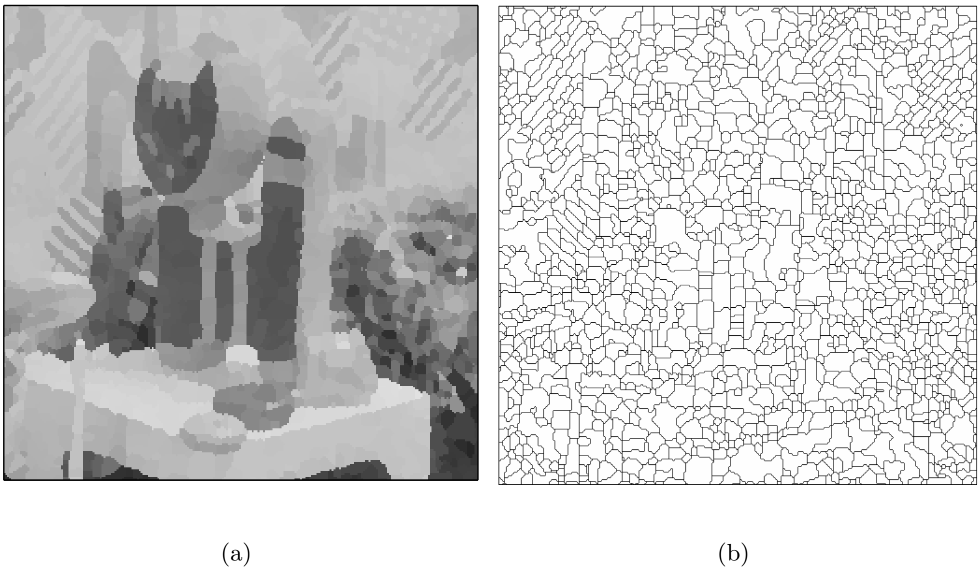
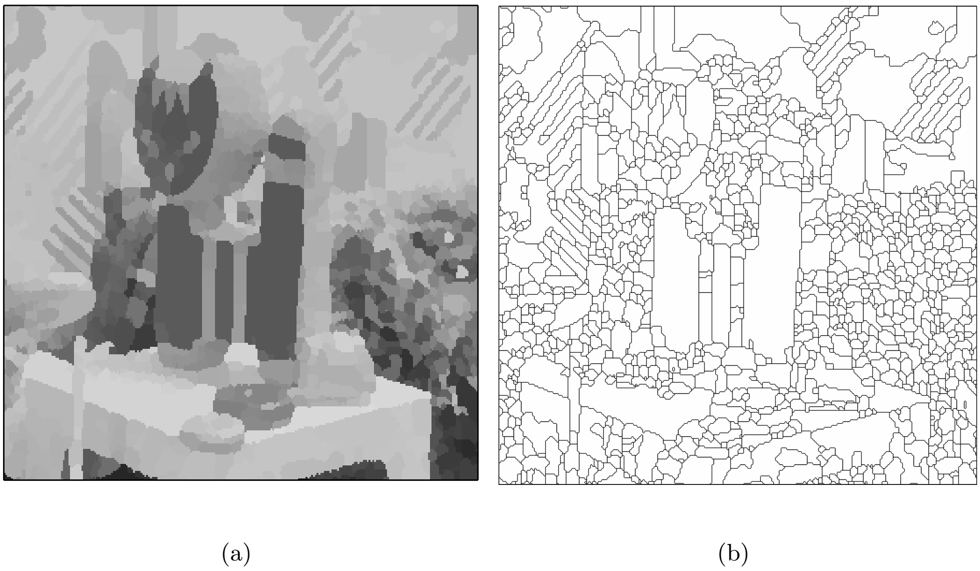


Figure 5.2: Sections of the mean replaced and outline images of the segmentation of the image of 'girl' using a linear Gaussian smoothing filter with $n_l = 12$ prior to gradient and watershed calculation.



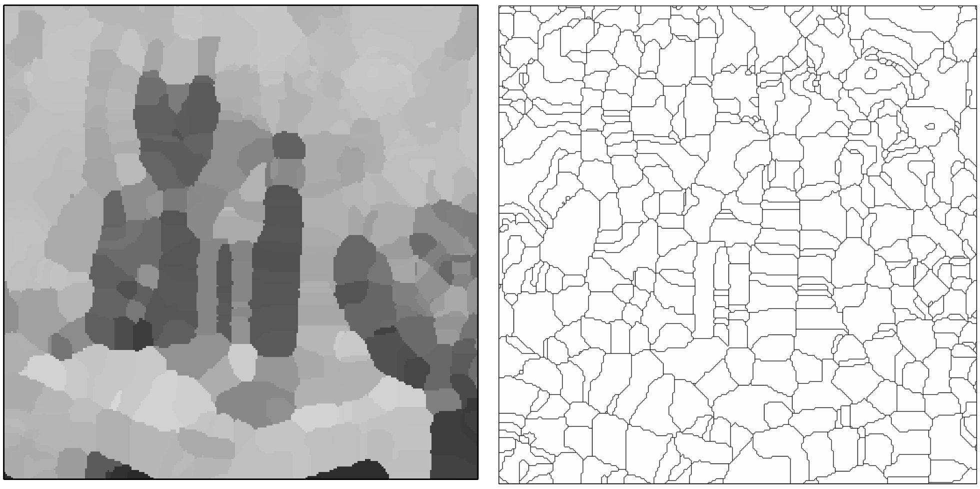
(a) (b)

Figure 5.3: Sections of the mean replaced and outline images of the segmentation of the image of ‘girl’ using a median smoothing filter with $n_m = 5$ prior to gradient and watershed calculation.



(a) (b)

Figure 5.4: Sections of the mean replaced and outline images of the segmentation of the image of ‘girl’ using the adaptive λ-max open-closing with $\lambda = 14$ prior to gradient and watershed calculation.



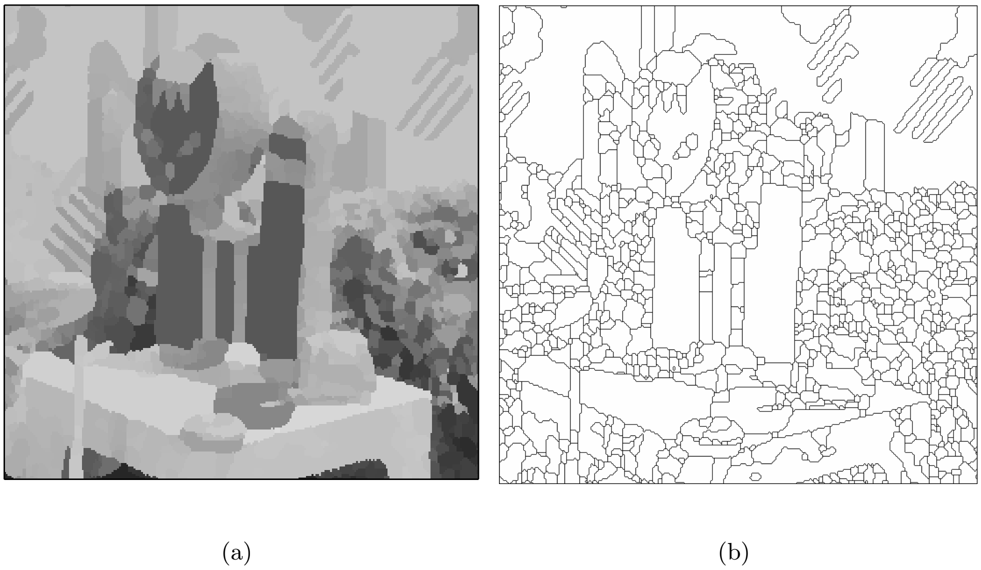
(a) (b)

Figure 5.5: Sections of the mean replaced and outline images of the segmentation of the image of 'girl' using the linear Gaussian filter prior to gradient and watershed calculation with $\alpha = 0.9$ in selection and a corresponding value of $n_l = 33$.

segmented. We can, as before introduce the weighting factor to the calculation of the total coding cost evaluating it according to Equation 4.10.

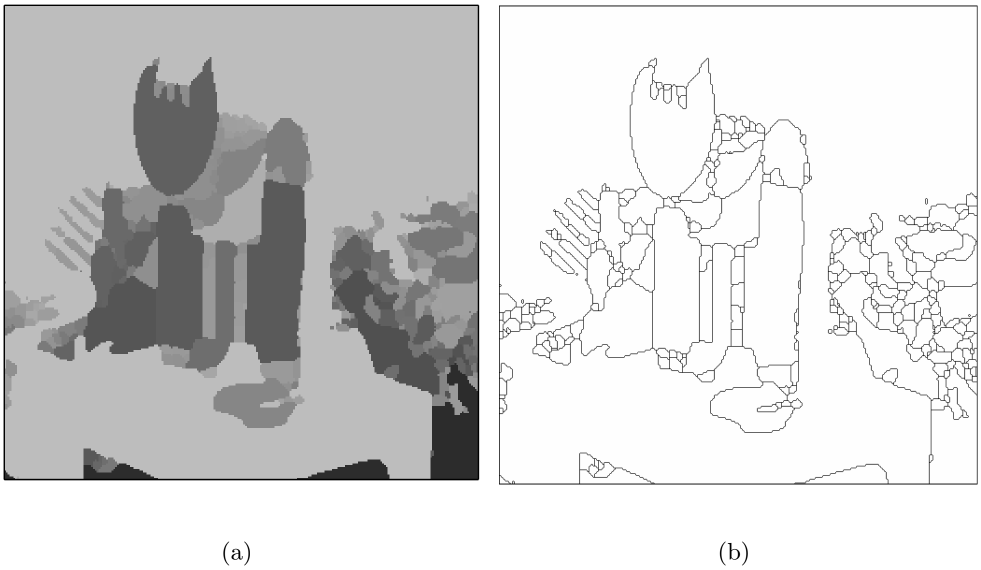
Figure 5.5 presents the mean replaced and outline images of the segmentation of 'girl' produced by smoothing the with a linear Gaussian filter selected with $\alpha = 0.9$. Though this may seem rather high when compared with those chosen previously, the images demonstrate that as we increase α the edges fail to correspond with object edges and the number of segments does not drop significantly. This is also demonstrated by the lower limit of the points of the scatter chart. Experimentation has shown that very similar results are achieved by increasing α for the median filter.

Figures 5.6 and 5.7 show sections of the segmentations resulting from the λ-max smoothing strategy using values of $\alpha = 0.6$ and 0.75 respectively. We note from these images that for a value of $\alpha = 0.6$ some regions of the image contain many segments while others relatively few. Increasing the value of α merges some of these segments but also results in single segments being associated with several objects. Experimentation with various images and values of α have shown that this problem



(a) (b)

Figure 5.6: Sections of the mean replaced and outline images of the segmentation of the image of ‘girl’ using the adaptive λ-max open-closing with $\alpha = 0.6$ in selection and a corresponding value of $\lambda = 20$.



(a) (b)

Figure 5.7: Sections of the mean replaced and outline images of the segmentation of the image of ‘girl’ using the adaptive λ-max open-closing with $\alpha = 0.75$ in selection and a corresponding value of $\lambda = 64$.

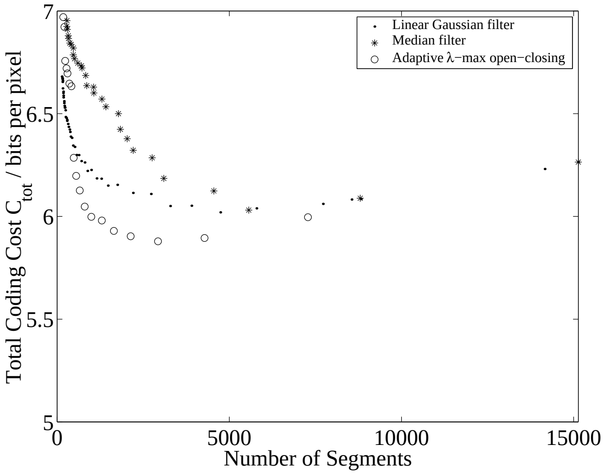


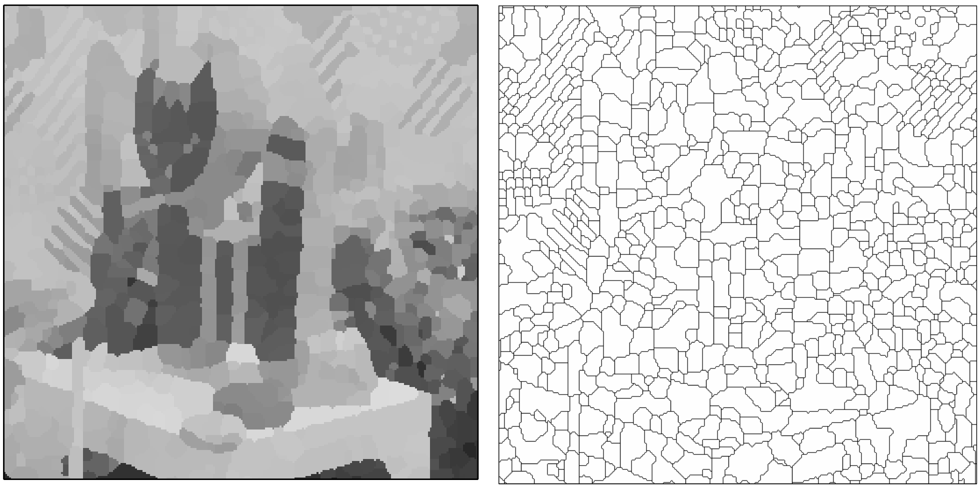
Figure 5.8: The total coding cost C_{tot} of the segmentations of the image 'girl' produced by smoothing the image gradient using linear Gaussian, median and adaptive λ-max open-closing prior to computation of the watersheds, plotted as a function of the number of segments.

is hard to overcome and results are not subjectively as pleasing as those obtained previously with either the adaptive λ-max and grey scale-merging operators or the coding cost minimising operator. And so we move to our final segmentation scheme.

5.2 Gradient Smoothing

The second scheme proposed here involves the same techniques of image smoothing applied in the previous section, but rather than use these for the pre-processing of the image, they are applied to the gradient image prior to the watershed transform.

Figure 5.8 shows a plot of the total coding cost C_{tot} as a function of the number segments for the segmentations produced by smoothing the gradient of the image of 'girl' prior to taking the watershed transform. In this case, the values of n_l, n_m and λ used to produce the segmentation range from 1 to 50. The results, presented as before in the form of a scatter chart, show that as with the image smoothing scheme, the adaptive λ-max open-closing has produced results with the lowest of the three coding costs. We also observe that the number of regions at which this minimum occurs is significantly lower than for the image smoothing scheme. The



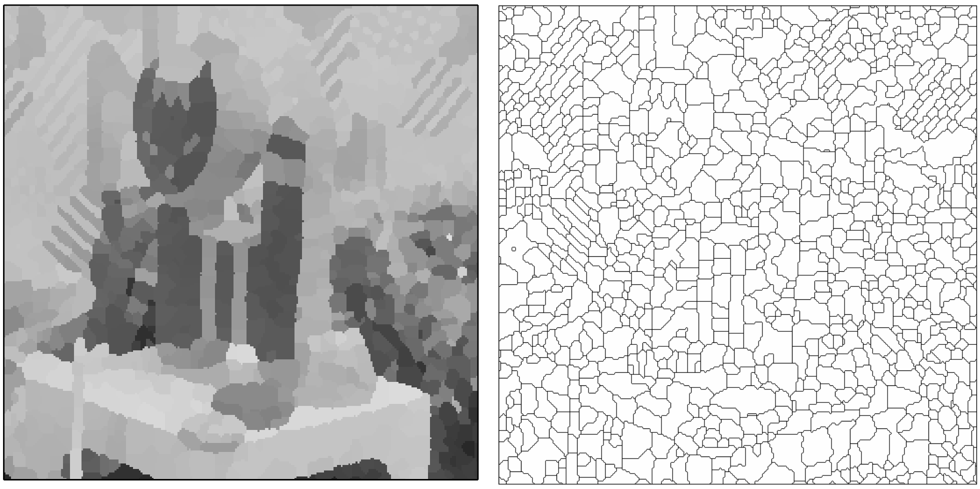
(a) (b)

Figure 5.9: Sections of the mean replaced and outline images of the segmentation of the image of ‘girl’ using a linear Gaussian smoothing filter with $n_l = 7$ on the gradient image prior to watershed calculation.

mean replaced and outline images for the linear Gaussian, median and adaptive λ-max operators are presented in Figures 5.9, 5.10 and 5.11 respectively.

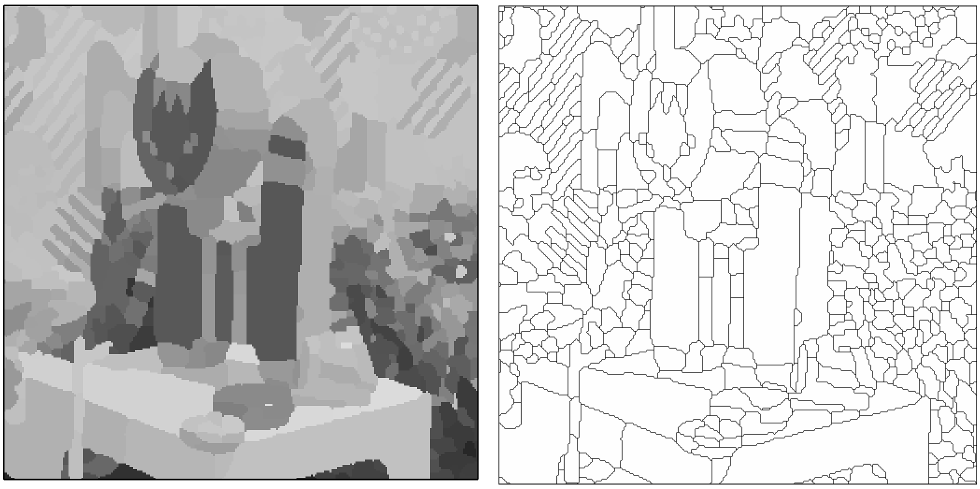
As before, the results for the linear and median filters are over-segmented and applying the weighted coding cost calculation with $\alpha > 0.5$ yields results in which the segment boundaries have arbitrary locations. The results for the adaptive λ-max however appear to meet our objectives and respond well to increasing values of α. Application of the weighted coding cost calculation to these segmentation results, for values of $\alpha = 0.6$ and 7.5, are presented in Figures 5.12 and 5.13.

The mean replaced and outline images for the segmentations resulting from the application of this scheme using $\alpha = 0.5,\ 0.6$ and 0.75, to the image set used in the previous chapter are presented in Appendix E and the values of un-weighted coding cost, segment numbers and λ are tabulated in Table 5.2.



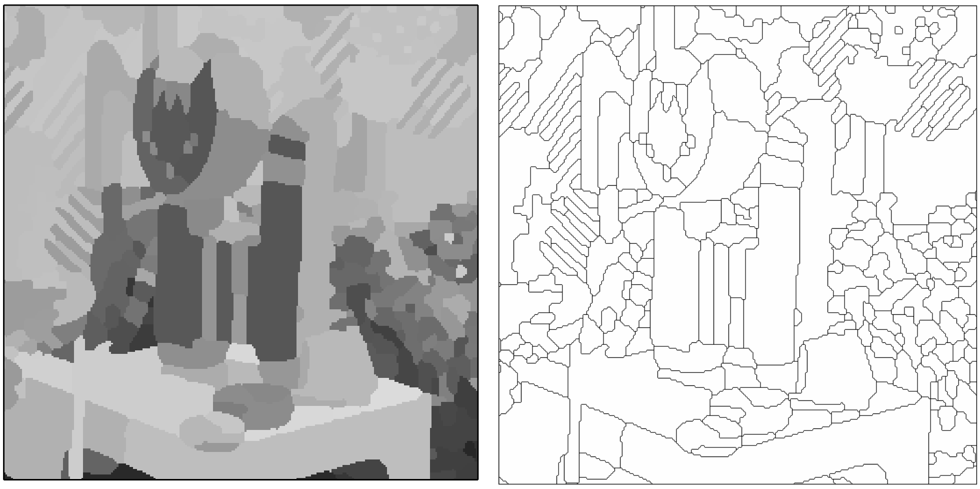
(a) (b)

Figure 5.10: Sections of the mean replaced and outline images of the segmentation of the image of 'girl' using a median smoothing filter with $n_m = 3$ on the gradient image prior watershed calculation.



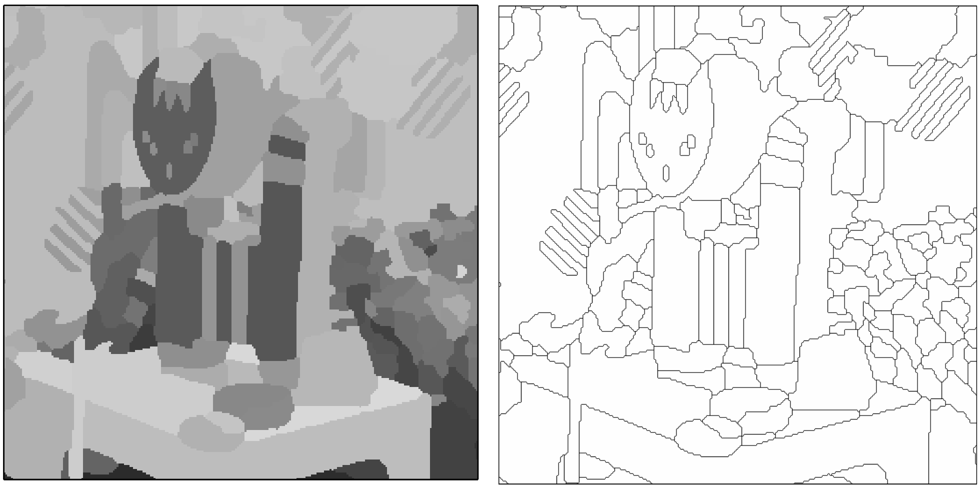
(a) (b)

Figure 5.11: Sections of the mean replaced and outline images of the segmentation of the image of 'girl' using the adaptive λ-max open-closing with $\lambda = 3$ on the gradient image prior to watershed calculation.



(a) (b)

Figure 5.12: Sections of the mean replaced and outline images of the segmentation of the image of 'girl' using the adaptive λ-max open-closing on the gradient image with $\alpha = 0.6$ in selection and a corresponding value of $\lambda = 5$.



(a) (b)

Figure 5.13: Sections of the mean replaced and outline images of the segmentation of the image of 'girl' using the adaptive λ-max open-closing on the gradient image with $\alpha = 0.75$ in selection and a corresponding value of $\lambda = 8$.

	No. of Segments	Coding Cost (C_{tot})	λ
'boat' (256 × 256)			
Original	50,501	6.384	-
$\alpha = 0.5$	252	6.389	6
$\alpha = 0.6$	252	6.389	6
$\alpha = 0.75$	44	6.641	34
'clown' (256 × 256)			
Original	45,792	7.119	-
$\alpha = 0.5$	563	6.317	2
$\alpha = 0.6$	206	6.396	6
$\alpha = 0.75$	35	6.598	23
'girl' (720 × 576)			
Original	316,108	7.227	-
$\alpha = 0.5$	2,929	5.879	3
$\alpha = 0.6$	1,646	5.930	5
$\alpha = 0.75$	799	6.048	8
'lenna' (256 × 256)			
Original	54,398	7.572	-
$\alpha = 0.5$	467	6.617	3
$\alpha = 0.6$	137	6.737	7
$\alpha = 0.75$	60	6.868	12
'toys' (256 × 256)			
Original	53,449	6.943	-
$\alpha = 0.5$	345	5.979	4
$\alpha = 0.6$	345	5.979	4
$\alpha = 0.75$	4	6.716	30
'tree' (256 × 256)			
Original	62,976	7.616	-
$\alpha = 0.5$	181	7.618	17
$\alpha = 0.6$	1	7.648	70
$\alpha = 0.75$	1	7.648	70

Table 5.1: A table of the number of segments, un-weighted coding cost and λ for the adaptive λ-max image smoothing scheme with $\alpha = 0.5,\ 0.6$ and 0.75 acting on various images. The resulting segmentations are presented in Appendix E.

The values of total coding cost presented in the table are higher than those due to the segmentations using either the adaptive λ-max and grey scale-merging or coding cost minimising schemes, though the results are subjectively pleasing in that they appear to meet our criteria for a segmentation. Finally, Figure 5.14 presents the complete mean replaced and outline images for the segmentation of the image of ‘girl’ using the gradient smoothing scheme corresponding to minimum weighted coding cost with α=0.75.

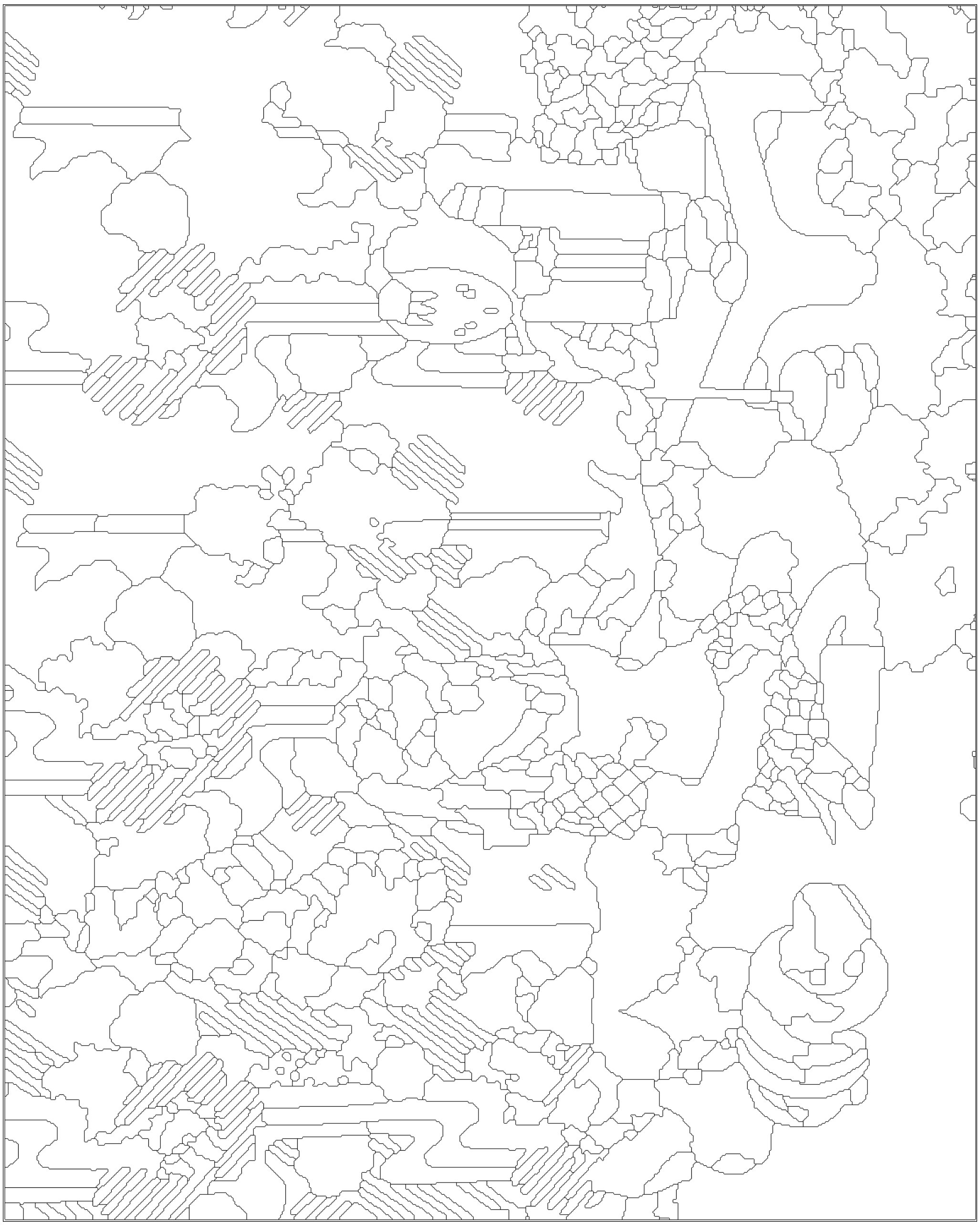
The final section of this chapter investigates the use of the adaptive λ-max open-closing / grey scale-merging operator and the gradient smoothing segmentation schemes in a novel application, namely noise removal using wavelets.

5.3 Image Segmentation for Improved Noise Reduction Using Wavelets

Image noise reduction is a highly active field of image processing with several applications including restoration and archiving [54]. In particular the use of the wavelet transform has been shown to be very powerful in this respect [55–59]. In this section we propose the use of the segmentation schemes developed in this thesis in an adaptation of the methods described in [56].

The basic principle of noise reduction using wavelets has three distinct stages. First, the image is decomposed into the wavelet components each of which is then subjected to a non-linear function often referred to as shrinkage, thresholding or coring. Once each component has been processed, the image is reconstructed by application of the inverse wavelet transform. The many variants of this method differ in their choice of wavelet transform, thresholding scheme and the adaptation of the thresholding parameters to the subject image. The decimated wavelet transform, though applicable to noise reduction using this scheme, has been shown to yield lower levels of noise reduction and higher levels of distortion than its un-decimated counterpart [58]. As such, it is the un-decimated wavelet transform which we shall use as the basis of this study.

(a)



(b)

Figure 5.14: The mean replaced image of 'girl' after application of the adaptive λ-max gradient smoothing scheme with $\alpha = 0.75$ (a) and the corresponding outline image (b).

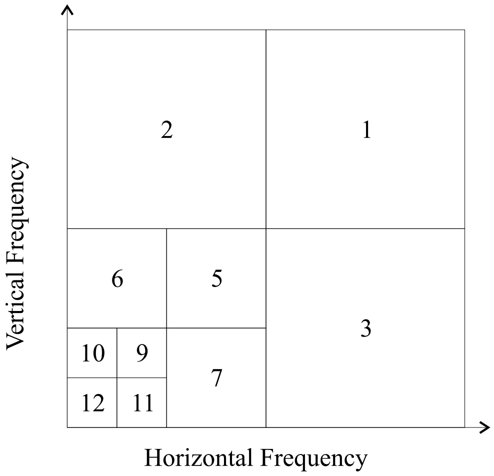


Figure 5.15: A schematic of the frequency decomposition of an image using wavelets.

5.3.1 The Un-decimated Discrete Wavelet Transform

Probably the most popular transform for signal analysis is the Fourier transform which attempts to decompose the signal as a possibly infinite summation of sinusoidal basis functions. These basis functions, by definition have infinite support and as such are unsuited to the decomposition of transient signals. Image functions, in which discontinuities proliferate have many, spatially localised or transient features and as such the use of the Fourier transform is often regarded as unsuitable [55]. The solution is to choose basis functions with finite support. One such decomposition technique is the wavelet transform.

The wavelet decomposition effectively decomposes the image using a bank of filters consisting of two basic designs, high pass H_{hp} and low pass H_{lp}. In the application of the wavelet transform to images, these must be applied in both horizontal and vertical directions. Each component then has an associated frequency band in both horizontal and vertical directions as depicted in Figure 5.15 each band (square) of which represents a component. Component 1, frequently referred to as the HH component is derived by filtering the image twice (once in each direction) with the filter H_{hp}^1. Component 2, the HL component has been filtered first with H_{lp}^1 in the horizontal direction and then H_{hp}^1 in the vertical. Component 3, the LH component is derived by first filtering the image with the H_{lp}^1 in the vertical direction followed by H_{hp}^1 in the horizontal. Component 4, being that associated with the bands comprising 5 through 12 in Figure 5.15 is the result of filtering the image in both horizontal and vertical directions using the filter H_{lp}^1 and is termed the LL component. In the decimated case, the resulting images are decimated or sub-sampled, while in the un-decimated case, zeros are inserted between the filters coefficients (hence the superscript notation) and the decimation is not performed. Components 6 to 8 (being that comprising bands 9 to 12) are derived in similar fashion, but the filter H_{lp}^1 has a zero interleaved between each of its coefficients, resulting in the filters H_{lp}^2 and H_{hp}^2. Calculation of the components 9 to 12 follows this pattern with the filters H_{lp}^3 and H_{hp}^3 with additional zeros inserted. The complete filter bank is depicted in Figure 5.16 in which the second subscript refers to the direction in which the filter is applied. Note that only three levels of decomposition are applied as it has been shown that up to three, the noise reduction improves dramatically but beyond this yields littel improvement [55].

Traditional schemes require a set of synthesis filters in the reconstruction stage however, in [55], the author proposes combining the analysis and synthesis filters allowing the components to be summed directly in the reconstruction stage. The author goes on to show that the properties of this scheme are better suited to noise reduction. The impulse response of the basic filters we use are given below.

$$\begin{aligned} H_{hp}^1 &= \frac{[1, 0, -9, 16, -9, 0, 1]}{32} \\ H_{lp}^1 &= \frac{[-1, 0, 9, 16, 9, 0, -1]}{32} \end{aligned} \tag{5.1}$$

Having described how the image can be decomposed into its wavelet components we now discuss the method of removing the noise from these components.

5.3.2 Noise Removal using Image Segmentation

In the removal of noise from the wavelet components we use a soft thresholding function of the following form in which $W^j(\mathbf{x})$ is the j^{th} wavelet component and θ is the thresholding parameter.

$$W_{\text{soft}} = f(W^j, \theta) = \begin{cases} \text{sign}[W^j][|W^j| - \theta] & |W^j| > \theta \\ 0 & |W^j| \leq \theta \end{cases} \tag{5.2}$$

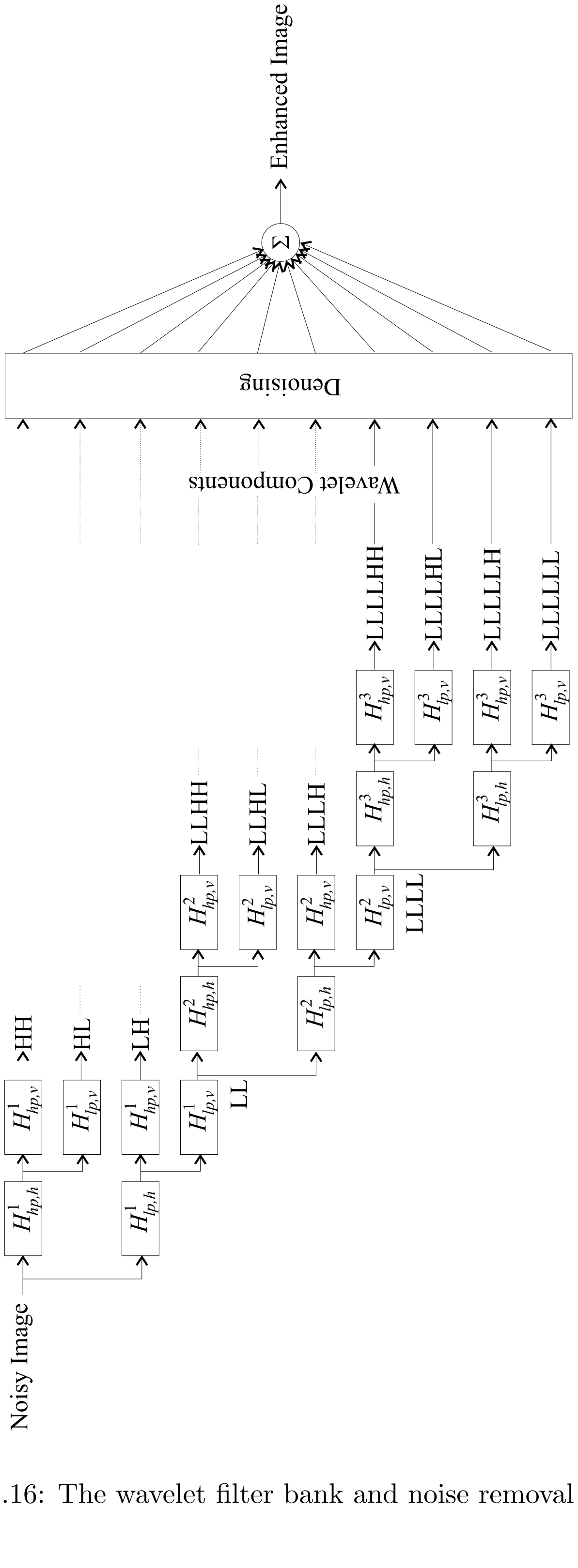


Figure 5.16: The wavelet filter bank and noise removal scheme.

In the method described in [56] the authors prescribe to each wavelet component a different threshold based upon the measured noise properties. In the first instance, the standard deviation of the noise $\hat{\sigma}_n$ is estimated using the median absolute deviation (MAD) of the HH component W_J, as described by Equation 5.3.

$$\hat{\sigma}_n = 1.483\text{med}|W_J| \tag{5.3}$$

The standard deviation of the noise free image in the j^{th} component is then estimated according to Equation 5.4.

$$\hat{\sigma}_{s,j} = \max\left\{0, \sqrt{\frac{1}{MN-1}\sum_{m,n} W_j}\right\} \tag{5.4}$$

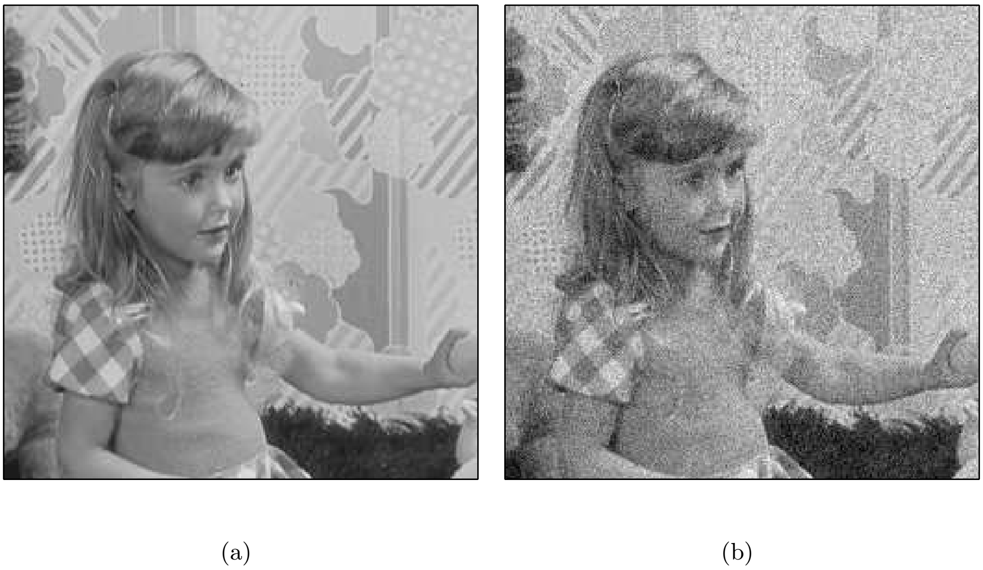
This approximation assumes that the noise is spatially uncorrelated. A suitable choice of the thresholding parameter θ is evaluated according to the following expression.

$$\theta = \frac{\hat{\sigma}_n^2}{\hat{\sigma}_{s,j}} \tag{5.5}$$

A large value of θ will suppress more noise, but also introduce more distortion that a low value and as such accurate estimation is critical. The authors of [56] go on to describe a scheme in which this method is applied adaptively by evaluating θ as a function of pixel context. The method proposed here can be regarded as an extension of this.

If the contaminating noise is assumed to have spatially stationary characteristics the signal to noise ratio (SNR) is not constant but varies with image content. As such by evaluating the thresholding value as a function of the local statistics we can apply more aggressive thresholding in regions where the SNR is low. The local statistics are estimated by evaluating $\hat{\sigma}_{s,j}$ for each segment of a segmentation. By using a segmentation in which the segment boundaries follow natural discontinuities in the image it is hoped that the noise reduction can be increased without the introduction of perturbing artefacts.

Specifically the adaptive λ-max gradient smoothing technique is used and the gain in signal to noise ratio evaluated for various values of λ. It should be noted that the segmentation techniques are applied to the noisy image as it is assumed that the original image, which we have contaminated artificially with white Gaussian noise



(a) (b)

Figure 5.17: A section of the original image of ‘girl’, before (a) and after (b) addition of white Gaussian noise of standard deviation $\sigma_n = 15$.

of standard deviation 15, is not known. Figure 5.17 shows the section of the image of ‘girl’ used throughout before and after contamination.

Figure 5.18 shows the noise reduction, calculated according to the gain in signal to noise ratio identical to that used in our study of impulsive noise removal (Equation B.1) but reiterated here in context.

$$\mathrm{SNR}_{\mathrm{gain}} = 10 \log_{10} \frac{\mathrm{MSE}[f(m,n), g(m,n)]}{\mathrm{MSE}[f(m,n), \Psi(g(m,n))]} \tag{5.6}$$

In which $f(m,n)$ is the original, pristine image, $g(m,n)$ the noisy image and Ψ refers to the noise removal scheme presented above. This clearly shows that using the segmentation has produced a significant increase in the signal to noise ratio over applying the scheme to the whole image, the increase for which is given by the point for which only one segment exists. The exact value is 6.9dB and occurs when there are 163 segments which, in terms of our segmentation scheme has a corresponding value of $\lambda = 12$. The segment outline and enhanced image corresponding to the maximum of Figure 5.18 are given in Figure 5.19. We note from the segmentation outline that the segmentation corresponding to the ‘best’ noise reduction is in fact an over-segmentation. It is suggested that methods of segmentation basing themselves on the statistical properties of the image, such as those proposed in [60] may produce

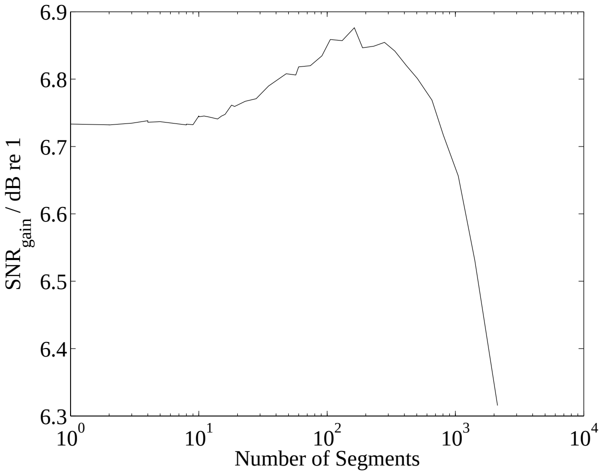


Figure 5.18: The gain in signal to noise ratio $\mathrm{SNR}_{\mathrm{gain}}$ as a function of the number of segments produced using the adaptive λ-max gradient smoothing segmentation scheme.

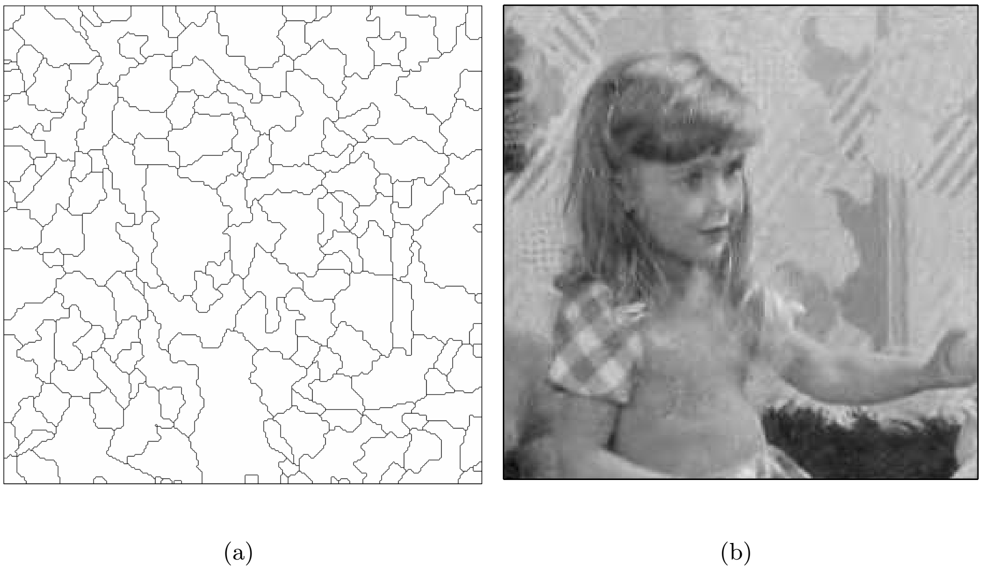


Figure 5.19: The segmentation outline (a) and enhanced image (b) corresponding the the best $\mathrm{SNR}_{\mathrm{gain}}$ obtained using the adaptive λ-max gradient smoothing segmentation scheme for the section of the image of ‘girl’.

	No. of Segments	$\mathrm{SNR}_{\mathrm{gain}}$ /dB ('best')	$\mathrm{SNR}_{\mathrm{gain}}$ /dB (Single Segment)	λ
'boat' (256 × 256)	645	5.419	4.792	5
'clown' (256 × 256)	615	6.942	6.189	5
'girl' (256 × 256)	163	6.876	6.456	12
'lenna' (256 × 256)	746	6.289	5.441	4
'toys' (256 × 256)	428	6.888	6.483	7
'tree' (256 × 256)	768	2.059	1.890	4

Table 5.2: A table of the number of segments, corresponding value of λ and the SNR gains for the 'best' and single segment cases. The resulting images are presented in Appendix F.

further improvement. Results of applying this noise reduction scheme to several images corrupted by white Gaussian noise with $\sigma_n = 15$ are presented in Appendix F. Values of the number of segments, λ and SNR gain for the 'best' noise reduction, along with the SNR gain achieved by the scheme without segmentation for these images are presented in Table 5.2.

We note from the table that an improvement in the SNR gain over the scheme without segmentation has been achieved without the introduction of undesirable artefacts in the enhanced images. Research into more suitable segmentation schemes for this application has not been conducted but is suggested for future work.

5.4 Summary

This chapter has introduced two schemes by which the watersheds tendency to over-segment can be indirectly controlled. In the first, the image is smoothed using the adaptive λ-max open-closing prior to the calculation of the gradient and the results are shown to compare favourably with he use of the linear Gaussian and median filters both in terms of our metric and the objectives of this thesis but it was noted that the number of segments may too high for many applications. Though the introduction of the weighted coding cost was shown to improve this, experimentation has shown that a suitable value of α is rather hard to obtain.

The second scheme also uses the adaptive λ-max open-closing as a smoothing function, but in this case it is applied to the gradient image. Once again results show improvement over the use of linear and median filters applied in the same way. Introduction of the weighted coding cost calculation was shown to allow tuning of this scheme producing results which meet the objectives of this thesis albeit with unweighted total coding cost values higher than those obtained in the previous chapter. Results for the application of this scheme to various images were presented.

Finally it was shown that the use of an image segmentation in the established method of noise removal using the wavelet decomposition improves the noise reduction in terms of the gain in signal to noise ratio.

Chapter 6

Conclusions

Image segmentation is a particularly active research topic within the field of image processing. The motivations behind the research are twofold. High-level, or complete segmentation aims to identify as segments, objects within the scene, while partial segmentation aims to segment the image into regions which are homogeneous with respect to some property such as grey level, colour or texture. Partial segmentations are often used as pre-processing steps to higher-level image processing tasks, including complete segmentation, allowing subsequent processing to be performed at a segment rather than pixel level, considerably reducing computation load and often, as in the case of motion estimation, improving results. It is because of the wide ranging applications of partial image segmentation that we choose them to form the basis of this research.

In order to give direction to our review of the numerous partial image segmentation techniques which are presented in the literature it was necessary to specify the desirable properties a segmentation. We chose to base these properties on the application of motion estimation though it is stressed that these properties are applicable to a number of image processing tasks. We recapitulate these properties here.

- The information required for the segmentation must be derived from the image intensity (grey scale) function.
- The technique must require no semantic knowledge of the scene or user inter-

action.

- Segments must comprise connected sets of pixels.
- Segment boundaries should correspond well with object boundaries.
- Computational load must be kept to a minimum.

With these criteria in mind, Chapter 2 categorised the established techniques of image segmentation into global, edge and region based methods. Global methods, basing their definition of segments on the image histogram, were shown to produce segments which were not connected and as such these techniques failed to meet our criteria.

Edge based methods aim to locate edges in the image through the processing of first and second order gradient estimates. It was demonstrated that first order gradient thresholding methods are highly sensitive to noise in the image leading to a trade off between locating non-existent edges and missing those which are present. A second draw back of these techniques is that they do not directly define closed contours from which connected regions may be derived, a problem overcome to some extent by second order methods.

Estimation of the second derivative of a Gaussian smoothed image via the Laplacian operator was shown to produce closed contours, but as the degree of smoothing was increased, so as to suppress detection of insignificant edges, the contours correlated less well with the edges of homogeneous regions. As such we terminated our investigation of edge based segmentation methods and proceeded with those which aim to identify segments segments directly.

Region based segmentation techniques can themselves be subdivided into the categories of splitting, split and merge, and region merging algorithms, all of which follow the same basic philosophy of defining segments based on the local homogeneity of the image. We also include under the title of region based methods, morphological and connected operators.

The techniques of splitting and split and merge, basing their analysis on a convenient structure referred to as the quadtree, suffer from the fact that segment boundaries are highly geometrical and therefore do not correspond well with the natural boundaries of homogeneous regions in the image. Placing stringent homogeneity criteria on the splitting stage and altering the criterion of the merging stage was demonstrated to improve this to some extent but in the limit it was observed that the split and merge technique equates to region merging.

The region merging techniques of single and centroid linkage region growing were introduced and demonstrated to produce relatively pleasing results but for the drawback of computational efficiency. This can be overcome with the use of connected operators and it is these which are singled out for further investigation.

The final section of Chapter 2 defines the watershed algorithm and demonstrates that its application to the gradient image produces many segments, some of which have boundaries which correlate very well with homogeneous regions. The one drawback relates again to the presence of noise in estimating the gradient and subsequently the technique produces large numbers of segments.

Of all the techniques reviewed, region growing, and in particular connected operators and the watershed transform demonstrate potential for techniques which meet out criteria and are as such developed in the following chapters.

Chapter 3 develops the techniques of morphological and connected operators for segmentation in the context of a unifying framework, being the lattice of partitions. In particular the image operators referred to as open-closings are shown to have properties which make them suitable for image segmentation. Classical morphological open-closings are introduced through the operations of erosion and dilation and it is demonstrated that the results are highly dependant on the shape and size of the structuring element used in these operations. Three so called adaptive methods given the titles geodesic transforms, attribute morphology and filters by reconstruction promise to overcome the constraints of the structuring element and are developed under the title of connected operators. Specifically, the area, λ-max and adaptive λ-max operators are described within the context of the level set decomposition. As a slight deviation, Appendix B demonstrates the application of several operators to the removal of impulsive noise from grey scale image. A novel adaptation of the classical morphological open-closing filter in which orthogonal structuring elements are used in the opening and closing stages is shown to perform better than using a square structuring element in terms of the increase in signal to noise ratio. The best increase however is given by the adaptive λ-max operator which significantly outperforms the traditional median filter.

Chapter 4 begins with the introduction of a novel metric for the objective evaluation of an arbitrary partial image segmentation based on the cost of coding the image without loss using said segmentation. It is demonstrated, using a simple block image, that the metric can be used for the selection of segmentation algorithm parameters by minimising the coding cost. This method is then applied to the area, λ-max and adaptive λ-max open-closing operators for the selection of an optimum value of area threshold and λ. Comparing segmentations with equal numbers of segments reveals that for a reasonable number the adaptive λ-max operator performs better than the other two in terms of the coding cost.

Inspection of the mean replaced and outline images of the segmentations resulting from the application of the open-closing operators shows regions consisting of many small segments, we refer to as transition zones. These transition zones prevent good correspondence of segment boundaries with the edges of homogeneous regions in the image. The development of two novel connected operators, the area- and grey scale-merging operators mitigate this effect considerably. Of these, the grey scale-merging operator yields the greatest reduction in coding cost as well as providing superior correlation of segment boundaries with the edges of homogeneous regions in the image.

A weighting factor was introduced to the metric penalising segmentations with high outline and means coding costs and this is shown to allow control over the number of segments present in the segmentation deemed ‘best’ in terms of the coding cost.

The final section of Chapter 4 develops a novel connected operator which aims to minimise the coding cost of the segmentation by iteratively merging segments if the resulting segmentation yields a drop in the coding cost. The result is a segmentation algorithm which performs better, in terms of the coding cost, than those previously analysed. The use of the weighted metric in the algorithm is once again shown to allow control over the number of segments produced.

Chapter 5 introduced two novel schemes for the assuagement of the watershed transforms tendency to over-segment the image. These take the form of applying connected operators, and specifically the adaptive λ-max open-closing, to either the image, as a pre-processing stage, or to the gradient image prior to computation of the watershed transform. The results of these non-linear smoothing operators are compared, in both schemes, with linear Gaussian and median filters.

It was demonstrated that the application adaptive λ-max operator to the image prior to gradient calculation produces segmentations with lower associated coding costs than either linear or median filters. The weighted metric was introduced to allow segmentations with fewer segments to correspond with the minimum coding cost but experimentation has shown that linear and median filters do not perform at all well under these conditions. The adaptive λ-max scheme performs better in terms of segment boundary location, but the segmentations contains regions with highly variable sizes.

In the application of the filters to the gradient image, the adaptive λ-max operator is once again shown to produce segmentations with lower coding costs than the others. Introduction of the weighted metric produces segmentations corresponding to the minimum coding cost with fewer segments, as before, but in the case of linear and median filters the segment boundaries do not correspond at all well with edges in the image. The minimal coding cost results for the adaptive λ-max operator using the weighted metric are very pleasing, the segments showing good correlation with significant regions in the image.

Finally, a novel, spatially adaptive variant of the popular wavelet based method of noise removal is introduced. The method uses the concept that if the noise is spatially uncorrelated, and the image function varies over the image, then the signal to noise ratio will also vary. As such application of a spatially varying threshold in the coring function applied to the wavelet components will improve the increase in signal to noise ratio. The statistics of the image and hence the coring function threshold are therefore estimated on a segment by segment basis, using the adaptive λ-max gradient smoothing segmentation scheme. The results demonstrate an improvement over methods in which the statistics are estimated over the entire image.

6.1 Further Work

Though work in the application of connected operators to image segmentation in general is highly recommended, a number of specific topics, identified throughout this thesis, are deemed worthy of special attention.

It was noted in the development of the segmentation metric presented in Chapter 4 that the simple compression scheme employed for the exploitation of spatial redundancy in the outline image may not be minimal. This implies that the entropy of the compressed image may still not represent the minimum cost of coding the segmentation outline which in turn effects the total coding cost. As such, methods which seek to exploit the spatial redundancy further should be investigated.

In applying the metric to the optimisation of segmentation algorithm variables, a large number of segmentations must be computed before the one giving minimal total coding cost can be identified. An in depth study of the nature of the coding cost surface and optimisation techniques based upon this study are recommended.

The final section of Chapter 4 introduced a connected operator which seeks to minimise the coding cost associated with the segmentation by merging regions provided the result has a lower coding cost. In so doing, the coding cost of the segmentations at every stage must be evaluated explicitly. The result is a very inefficient algorithm. If an implicit relationship between the coding cost and the regions which are proposed for merging can be derived the computational efficiency will almost certainly be increased. This will allow for the introduction of more constraints on the merging criteria, such as grey level difference.

The final section of Chapter 5 introduced a novel, spatially adaptive variant of the wavelet de-noising technique in which the statistics of the image are evaluated on a segment by segment basis. Though the segmentations achieved using the gradient smoothing watershed scheme presented earlier in the chapter were shown to yield an improvement in the increase in signal to noise ratio it is suggested that a partial segmentation scheme in which an estimate of the signal to noise ratio is used as a homogeneity criterion may produce further improvement.

Appendix A

Fundamentals of Set Theory

A.1 Introduction

This Appendix serves as a glossary for the symbols and terms of set theory which are used extensively throughout this report. Much of this work can be found in [61] and [62].

A.2 Sets, Elements and Set Equality

A set is a collection of definitive distinct object of our perception or of our thought, which are called elements of the set[61]. A *null* or *empty set* contains no elements and is given the symbol $\emptyset$. $\emptyset$

The examples throughout this section concentrate on sets of integers as this leads naturally to the representation of digital images, however the operations may be applied equally well to any element type. A set A may consist for example of the integers 1,2 3 and 4 and would be written enclosed in braces and separated by commas thus:

$$A = \{1, 2, 3, 4\}$$

The integers 1,2,3 and 4 are elements of the set A, this inclusion can be expressed as $n \in A$ and reads; "n is and element of A", where $n = 1, 2, 3, 4$. Conversely, the integer 6 is not an element of A, this is expressed as $6 \notin A$. The symbol $\forall$ reads "for all", for example, "for all $a \in A$" is written $\forall a \in A$. Taking as the domain the set of all integers, $\mathbb{Z}$, the set of integers $m \in \mathbb{Z}$ for which $m \notin A$ is called the complement of the set A and is denoted by A^c.

$\in$

$\notin$

^{c}

The set B is defined as:

$$B = \{4, 3, 2, 1\}$$

It is clear that A and B contain the same elements and as such the sets are said to be equal and we write $A = B$. The equality relation for non-empty sets is reflexive, symmetric and transitive, i.e.

$=$

1. $A = A$
2. if $A = B$ then $B = A$
3. if $A = B$ and $B = C$ then $A = C$

If the elements of a set A such that $a \in A$, are defined as having some property $P(a)$, the notation $A = \{a|P(a)\}$, often called set-builder notation is used. The set A defined above could equally have been defined as:

$\{|\}$

$$A = \{a \in \mathbb{Z} \quad | \quad a \geq 1 \text{ and } a \leq 4\}$$

where $\mathbb{Z}$ is the set of all real integers.

A.3 Subsets, Union and Intersection

A set C is said to be a *proper subset* of A, denoted by $C \subseteq A$ if every element of B is also an element of A but not every element of A is an element of B. For example the set $C = \{1, 2, 3\}$ is a proper subset of the set A defined in the previous examples. A set C is said to be an *improper subset* of a set A, denoted $C \subset A$ if all the elements of C appear in A and all the elements of A appear in C, thus C=A. A set A is said to be a *superset* of C, denoted $A \supseteq C$, if all the elements of C are also

$\subseteq$

$\subset$

$\supseteq$

elements of A but not all the elements of A are elements of C. Finally, the empty set, $\emptyset$, is a subset of all sets.

The union of two sets, A and B, denoted by $A \cup B$, is the set of elements belonging to *either* of the two sets. Consider the sets $A = \{1, 2, 3, 4\}$ and $B = \{3, 4, 5, 6\}$. The union of the sets is given by: $\cup$

$$A \cup B = \{1, 2, 3, 4, 5, 6\}$$

The intersection of two sets, A and B, denoted by $A \cap B$ is the set of elements belonging to *both* sets, A and B. The intersection of the sets A and B used in the previous example is given by: $\cap$

$$A \cap B = \{3, 4\}$$

If two sets have no similar elements, their intersection is the empty set and the sets are said to be *disjoint*.

The operations of union and intersection obey the commutative, associative and distributive laws of standard algebra i.e.

1. Commutative:

$$\begin{aligned} A \cup B &= B \cup A \\ A \cap B &= B \cap A \end{aligned}$$

2. Associative:

$$\begin{aligned} (A \cup B) \cup C &= A \cup (B \cup C) \\ (A \cap B) \cap C &= A \cap (B \cap C) \end{aligned}$$

3. Distributive:

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

A.4 Set Subtraction, Cartesian Product, Functions and Mapping

The subtraction of, or difference between, two sets A and B is given the notation $A \setminus B$ and results in the set containing those elements which belong to the set A $\setminus$ but not set B. It can be defined formally using the intersection and complement operations thus:

$$A \setminus B = A \cap B^c \tag{A.1}$$

The cartesian product of two sets A and B takes the elements $a \in A$, and $b \in B$ and returns the set of couples, (a, b). More formally:

$$A \times B = \{(a, b) \quad | \quad a \in A \text{ and } b \in B\} \tag{A.2}$$

For example, let $A = 1, 2, 3$ and $B = 4, 5, 6$, the cartesian product is given by: $\times$

$$A \times B = \{(1, 4), (2, 4), (3, 4), (1, 5), (2, 5), (3, 5), (1, 6), (2, 6), (3, 6)\}$$

A function f taking as its domain the set A and mapping it onto a codomain B, is a relation between the sets A and B with the property that for each $a \in A$, there exists only one ordered pair $(a, b) \in f$ where $b \in B$. The mapping is written as $f : A \longrightarrow B$ and the relation $(a, b) \in f$ is more usually expressed in function notation $:\longrightarrow$ as $f(a) = b$. The range of the function is given by $f[A] = \{f(a) \quad | \quad a \in A\}$. For example, the addition of two positive integers (belonging to the set of all positive integers, $\mathbb{Z}^+$) can be expressed as the mapping of the Cartesian product of two sets of all positive integers onto the set of positive integers i.e. $+ : (\mathbb{Z}^+ \times \mathbb{Z}^+) \rightarrow \mathbb{Z}^+$. This can be thought of as the mapping of the matrix of all integer couples onto the vector of all integers thus:

$$\begin{bmatrix} (0,0) & (1,0) & (2,0) & \cdots \\ (0,1) & (1,1) & (2,1) & \cdots \\ (0,2) & (1,2) & (2,2) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & 1 & 2 & \cdots \end{bmatrix}$$

The action of the addition function of the couple $(3, 6) \in (\mathbb{Z}^+ \times \mathbb{Z}^+)$ can be written in function notation as $+((3, 6)) = 9$ or in set notation as $((3, 6), 9) \in +$.

We now have at our disposal the tools necessary to develop morphological operations but must first describe the representation of images by sets and how the operations thus far described effect images.

A.5 Images as Sets

The types of 'images' to which morphological operations may be applied range from the most basic binary images, with a two dimensional domain and binary range, to colour image sequences being defined over up to six dimensions. The most convenient space in which to define images is that of Euclid. Euclidean space of N dimensions, referred to as Euclidean N-space is given the symbol E^N and is the common space in which the Cartesian co-ordinate system is defined. The digital equivalent of Euclidean N-space is defined only at integer co-ordinates and is given the symbol $\mathbb{Z}^N$. While the focus of this report is images defined in the digital domain, many of the operations apply where the space is both continuous and digital, and therefore no distinction is made and the symbol E^N is used. We have seen, in the examples of preceding sections that sets can consist of integer elements. The representation of digital[1] images by sets requires that each of the elements is itself an ordered set of integers referred to as an N-tuple, where N is the number of elements in the set and must be greater the two. An N-tuple is a couple for $N = 2$, a triple for $N = 3$, a quad for $N = 4$ and so on. Couples for example can represent the cartesian cross product of the two Euclidean space dimensions used as the domain of still images, $(0, 0)$, $(1, 0)$ etc. as seen in the example of set functions.

A.5.1 Set Operations on Binary Images

Binary images are represented using a set of couples, the values of which represent the cartesian co-ordinates of foreground[2] pixels. Since, by definition a binary image can be either foreground or background, the set of all foreground pixels constitutes

[1] The use of the term 'digital' images implies that the image is defined in digital space and since images in continuous space are not discussed, shall hereafter be assumed.

[2] Throughout this report, foreground shall be given the value of 1 and represented by black regions, while background will be given the value zero and represented by white regions.

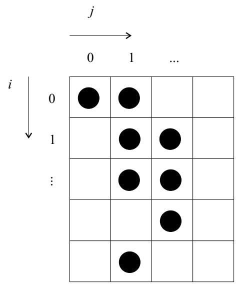


Figure A.1: A simple binary image with foreground pixels denoted by black dots.

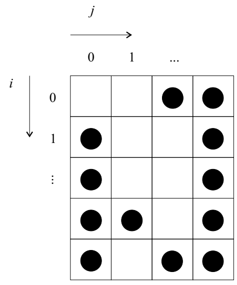


Figure A.2: The complement of the binary image given in figure A.1.

a complete description of the image, the extent of the background being unbounded. For example, figure A.1 shows a binary image whose corresponding set, A, is given by:

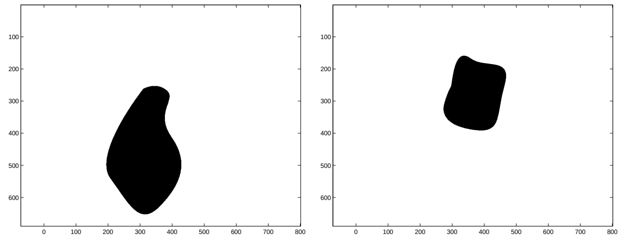
$$A = \{(0,0)\,(0,1)\,(1,1)\,(1,2)\,(2,1)\,(2,2)\,(3,2)\,(4,1)\}$$

The unbounded image background becomes a problem when we wish to take the set complement, that is the set containing all the background pixels. The complement of the set given above, for the domain shown in figure A.1 is given by the following;

$$A^c = \{(0,2)\,(0,3)\,(1,0)\,(1,3)\,(2,0)\,(2,3)\,(3,0)\,(3,1)\,(3,3)\,(4,0)\,(4,2)\,(4,3)\}$$

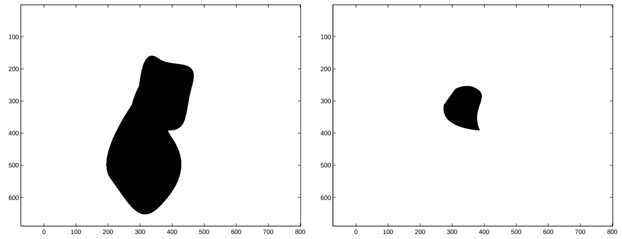
The image corresponding to the set given above is shown in figure A.2, note that is is simply the inverse of the original image.

Consider the two binary images given in figure A.3(a) and A.3(b), let us denote the corresponding sets as A and B respectively. These two sets are not disjoint since they have overlapping regions but neither is a subset of the other owing to the fact that no one set entirely encloses the other. The set union, $A \cup B$, and intersection,



(a) (b)

Figure A.3: Two binary images



(a) (b)

Figure A.4: The union (a) and intersection (b) of the two binary Images given in figure A.3.

$A \cap B$, of the two sets are show in figures A.4(a) and A.4(b) respectively. Note that the intersection operation results in an image whose foreground area is smaller than either A or B, while in the image resulting from the union operation the foreground area is greater than in either of the initial sets.

Another set operation relevant to the manipulation of image sets is that of translation. The translation of a set A, by a vector x, denoted by A_x is given by:

$$A_x = \{a + x \quad | \quad a \in A\} \tag{A.3}$$

The effect of this on a binary image set is that the image is shifted by the vector x.

A.5.2 Set Operations on Grey Scale Images

Increasing the dimension of the space to E^3, such that each element of the set representing the image is a triple allows the representation of three types of image, binary solids, binary image sequences and grey scale images. For binary solids, the third integer represents the position of the foreground pixel in the third Cartesian dimension, for sequences it represents time and for grey scale images the grey value.

Grey scale volumes or sequences can be represented using quad elements, the first three of which represent position or position plus time and the fourth the grey value.

When defining the set operations for grey scale still images, it is common for the triple to be split up into two components, $\mathbf{x}$, a vector describing the position of the pixel in the image, and y, the third element of the triple describing the grey value.

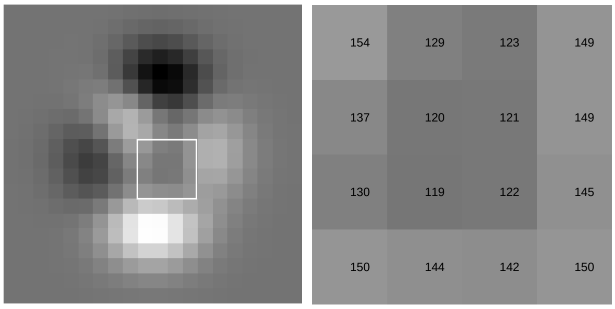
The image given in figure A.5 is a grey scale image created using Matlabs 'peaks' function. The set f, corresponding to the highlighted section is as follows;

$$\begin{aligned} f = \{\ldots &(9,9,154),(9,10,129),(9,11,123),(9,12,149),\ldots \\ \ldots &(10,9,137),(10,10,120),(10,11,121),(10,12,149),\ldots \\ \ldots &(11,9,130),(11,10,119),(11,11,122),(11,12,145),\ldots \\ \ldots &(12,9,150),(12,10,144),(12,11,142),(12,12,150),\ldots\} \end{aligned}$$

The complement of a grey scale set is basically the inverted (black for white) image. The practical implementation for this will depend on the system employed. In the example above, the 'peaks' image has an 8-bit colour range meaning that it can only take vales in the range 0 (black) to 255 (white). In general, The complement of the image is given by subtracting the grey value for each pixel from the largest available value i.e 255. In general, the maximum value an image described using an N-bit range can attain is $2^N - 1$. The set operation for grey scale complementation in the general case is given below.

$$f^c = \left\{(2^N - 1) - y \mid \forall(\mathbf{x}, y) \in B\right\} \tag{A.4}$$

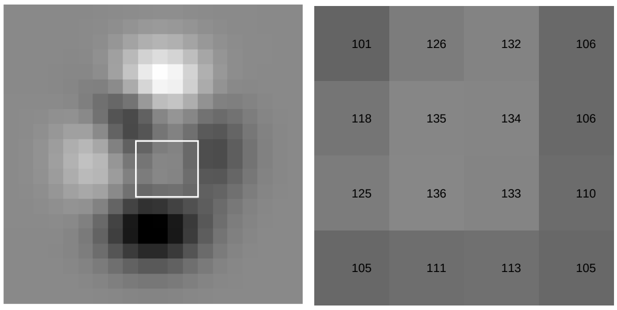
The set complement for the image 'peaks' is given along with the enlarged and labelled section in figure A.5.2.



154	129	123	149
137	120	121	149
130	119	122	145
150	144	142	150

(a) (b)

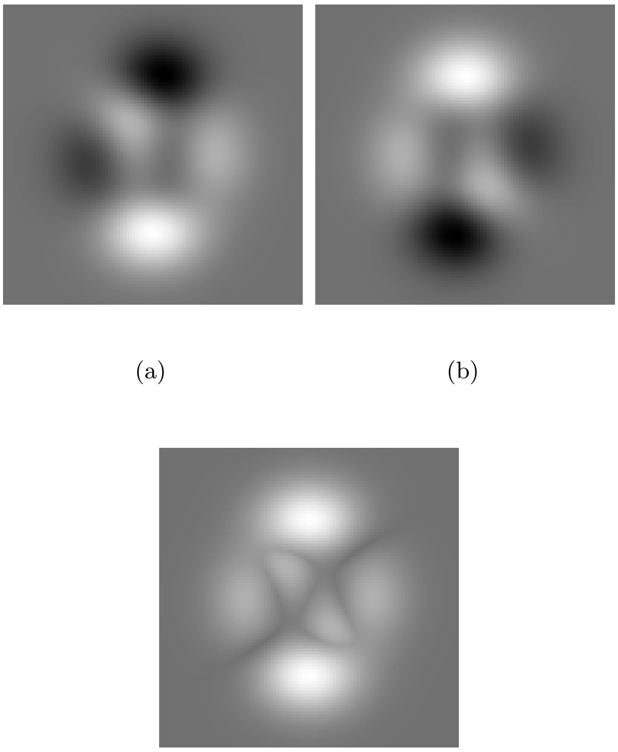
Figure A.5: Grey scale image of ‘peaks’ (a) and a enlarged and labelled section (highlighted) thereof (b).



101	126	132	106
118	135	134	106
125	136	133	110
105	111	113	105

(a) (b)

Figure A.6: The complement of the image ‘peaks’ (a), and the enlarged and labelled section thereof (b).



(a) (b)

(c)

Figure A.7: A high resolution image of 'peaks' (a), a 180° rotation thereof, (b), and the union of the two images (c).

The set theoretic operations of union and intersection are also defined for grey scale images. The union of two images f and g is given by the maximum value of either image at each pixel location. The set operation can be defined as follows;

$$f \cup g = \{\max\{y_f, y_g\} \mid \forall(\mathbf{x}, y_f) \in f \text{ and } (\mathbf{x}, y_g) \in g\} \tag{A.5}$$

Figure A.7 shows a higher resolution image of 'peaks' used in the previous example, a 180° rotation thereof and the union of the two images.

Correspondingly, the intersection of the two images is achieved by taking the minimum grey value achieved by either image at each pixel. The set theoretic operation can be defined as:

$$f \cap g = \{\min\{y_f, y_g\} \mid \forall(\mathbf{x}, y_f) \in f \text{ and } (\mathbf{x}, y_g) \in g\} \tag{A.6}$$

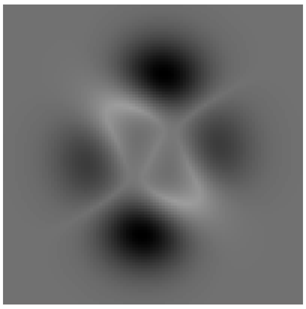


Figure A.8: The intersection of the images of ‘peaks’ given in figures A.7(a) and A.7(b).

Figure A.8 shows the result of performing the intersection operation on the original and rotated images of ‘peaks’ given in figures A.7(a) and A.7(b).

A.6 Summary

This appendix has introduced the basic concepts of a set, how they can be defined explicitly or conditionally and how sets relate to each other in terms of subsets, supersets and disjoint sets. The various operations required for morphology have been introduced including union, intersection and complementation and the concept of a set representing a two dimensional plane has been introduced. This has been developed into the representation of both binary and grey scale images using sets. In general, binary images defined over n dimensions can be described completely by a set of N-tuples of length n, and grey scale images in n dimensions can be described completely using an N-tuple of length $n + 1$, the first n of which describe position and possibly time and are often revered to as a vector ($\mathbf{x}$), leaving the $(n + 1)^{\text{th}}$ for the grey value, referred to throughout this report as y. The set operations of complementation, union, intersection and translation have been applied to example images.

Appendix B

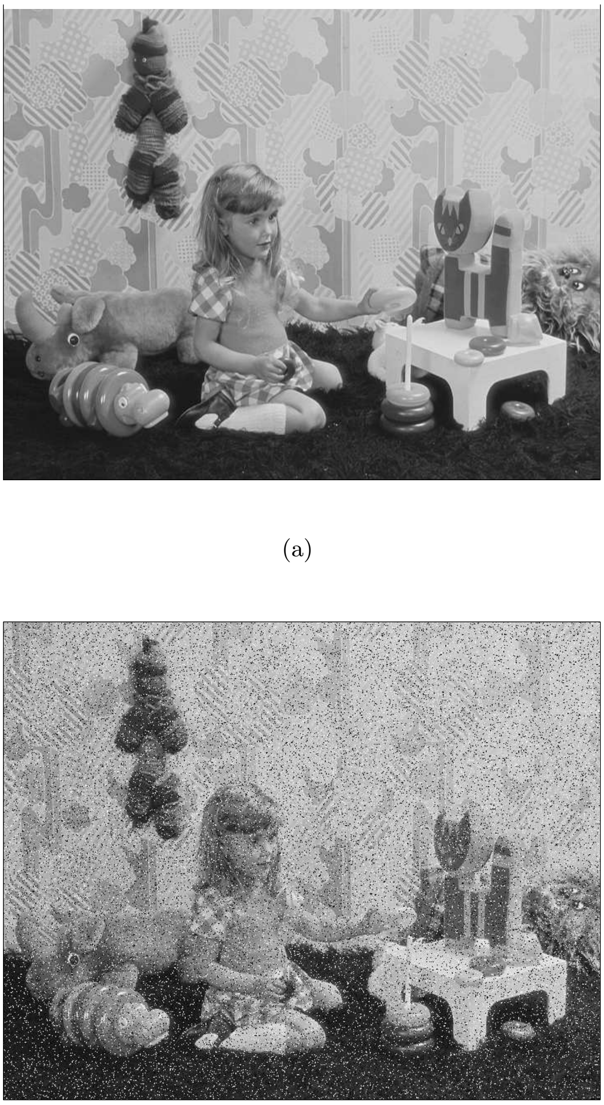
Impulsive Noise Removal Using Morphological and Connected Operators

It is well established in image processing that the removal of impulsive noise, unlike additive noise is not well suited to linear filters. The traditional method for the removal of such noise is the median filter [1]. In this section we compare the performance of the median filter with several strategies based upon the operators described in this chapter, namely morphological open-closing, area open-closing, the λ-max / λ-min or λ-max open-closing and the adaptive λ-max / λ-min or adaptive λ-max open-closing operators.

Figure B.1(a) shows the original image of 'girl' and Figure B.1(b) the same image after 10% of the images pixels have been currupted by salt and pepper noise in that they have been set to either black (0) or white (in this 8-bit example 255). The performance of the filtering strategies will be presented in terms of the increase in signal to noise ratio $\mathrm{SNR}_{\mathrm{gain}}$, calculated according to the following expression:

$$\mathrm{SNR}_{\mathrm{gain}} = 10\log_{10}\frac{\mathrm{MSE}[f(\mathbf{x}),g(\mathbf{x})]}{\mathrm{MSE}[f(\mathbf{x}),\Psi(g(\mathbf{x}))]} \tag{B.1}$$

Where $f(\mathbf{x})$ is the original image function, $g(\mathbf{x})$ the noisy image and Ψ the noise



(a)

(b)

Figure B.1: The original image of ‘girl’ (a) and the same image of corrupted by 10% impulsive noise (b).

reduction operator. MSE, the mean square error is calculated by the following:

$$\mathrm{MSE}[f(n,m),g(n,m)] = \frac{1}{NM}\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}[f(n,m)-g(n,m)]^2 \qquad \text{(B.2)}$$

where N and M are the horizontal and vertical dimensions of the image respectively. Using this measure of noise reduction we reward the removal of noise whilst penalising distortion that is necessarily introduced in any filtering process.

The median filter replaces the grey level of each pixel with the median grey level in the specified neighbourhood of said pixel. Taking for each pixel the central value of the set of ordered values, the filter has the effect of smoothing the image whilst preserving the 'sharpness' of the edges. In this study, a median filter of size n will imply that a square neighbourhood of size $n \times n$ has been used. Note also that only odd values of n have been used as neighbourhoods of even dimensions do not permit symmetry. A plot of $\mathrm{SNR}_{\mathrm{gain}}$ as a function of size is given in Figure B.2 from which it is clear that the median filter is most effective for this particular image when the median is taken over a 3×3 neighbourhood giving an increase of 17.8dB. Larger neighbourhoods introduce distortion artefacts and as such limit $\mathrm{SNR}_{\mathrm{gain}}$. Figure B.3 shows the noisy image of 'girl' after median filtering using a 3×3 neighbourhood from which it is clear that most of the noise has been successfully removed, but also fine textures such as the fur of the stuffed toys have lost some of their definition.

The morphological open-closings used will be of two types, the first using a square structuring element as in the median filter, but the second will use orthogonal line structuring elements for the opening and closing stages; a novel adaptation which significantly improves the noise reduction. In order to analyse the effect of morphological filter on the noisy image, we concentrate on the section given in Figure B.4(a). A section of the noisy image after opening with a 3×3 square structuring element is given in figure B.4(b) in which we note that although the 'salt' noise, consisting of regional maxima, has been removed, other regional maxima relating to image information have also been removed and the 'pepper' noise has increased in size. This is due to the fact that in the erosion not only are regional maxima shrunk in size, but by duality regional minima are enlarged and the pixels between closely spaced 'pepper' pixels are given the grey level 0. Once these bridges have been formed it is not possible for the dilation operation to remove them and we are effectively throwing away the information present in the original image. The result

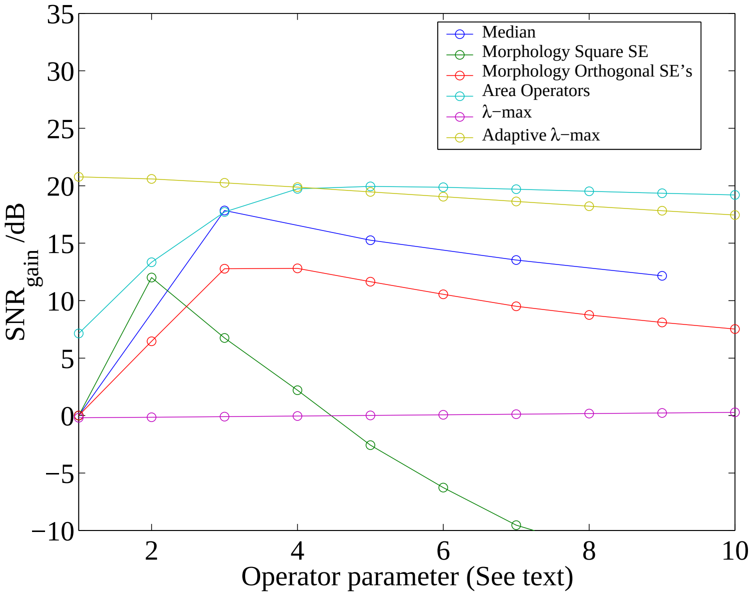


Figure B.2: The increase in signal to noise ratio $\mathrm{SNR}_{\mathrm{gain}}$ as a function of neighbourhood, structuring element size, area or λ.

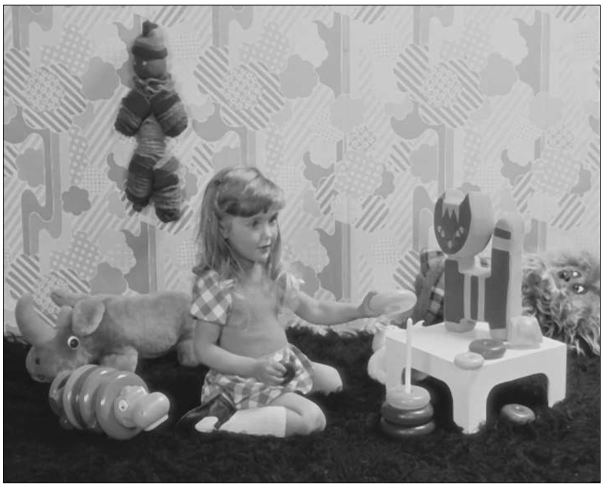
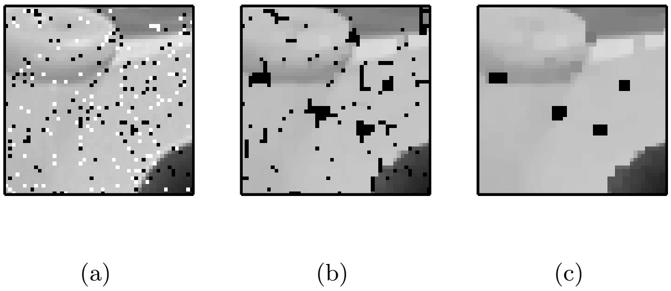
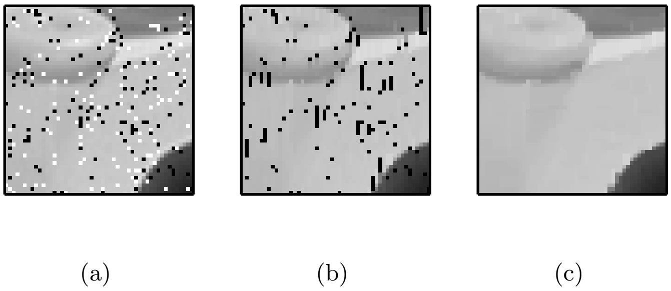


Figure B.3: The noisy image of 'girl' presented in Figure B.1(b) after median filtering with a 3×3 neighbourhood.



(a) (b) (c)

Figure B.4: An enlarged section of the corrupted image of ‘girl’ (a), the opening (b) and open-closing (c) thereof.



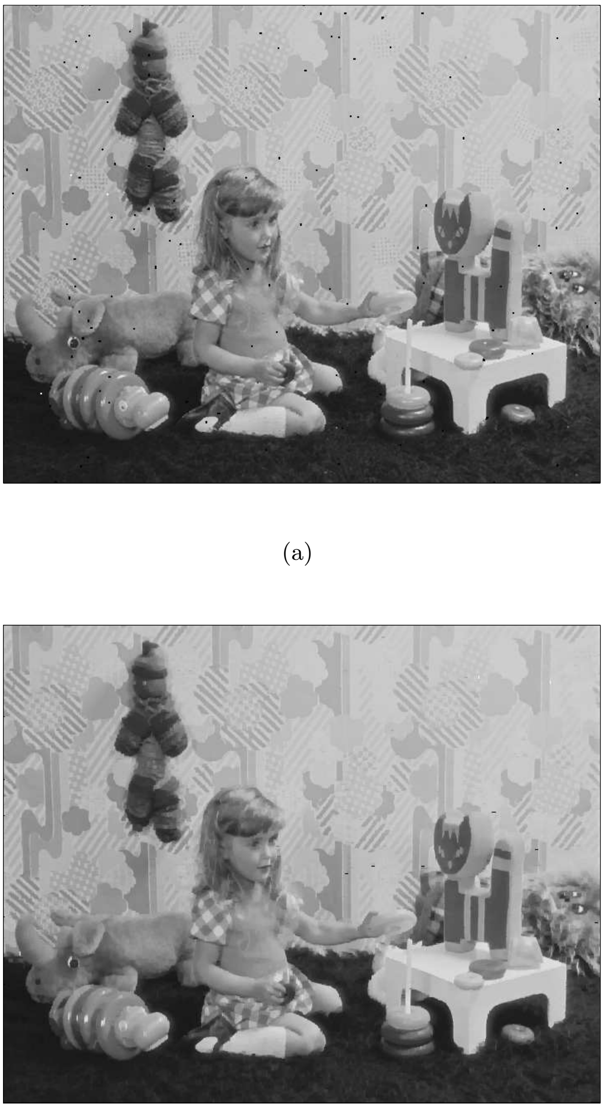
(a) (b) (c)

Figure B.5: An enlarged section of the corrupted image of ‘girl’ (a), the opening thereof by a 1×3 structuring element and (b) and open-closing with orthogonal line elements of length 3 (c).

is that in closing the image with a 3×3 structuring element the ‘pepper’ noise made larger by the opening cannot be removed and in some cases is made worse (Figure B.4(c)). Note that a similar effect will result in the close-opening of the image, but it is the ‘salt’ noise which will remain. One method of improving the removal of impulsive noise is to use orthogonal line structuring elements. Figure B.5(b) shows the result of opening the same section of the image ‘girl’ as in the previous example with a structuring element of size 1×3. Note that all the ‘salt’ noise has been removed as in the previous example but also that the ‘pepper’ noise has not been enlarged in the horizontal direction and thus can in the most part be easily removed using a structuring element of dimensions 3×1 as in figure B.5(c). The increase in signal to noise ratio as a function of structuring element size n, describing structuring elements of size $n \times n$ in the square case and line structuring elements of length n in the orthogonal case, is given on the same plot as the results for the median filter in Figure B.2. It is clear from these traces that orthogonal structuring element strategy outperforms the square, the square technique having a maximum increase of 12.0dB using a 2 × 2 structuring element and the orthogonal strategy resulting in an increase of 12.8dB using structuring elements of length 4. However, it is clear from Figure B.2 that neither morphological strategy performs better than the median filter. Figure B.6 shows the noisy image of 'girl' after morphological filtering with a square structuring element of size 2 and orthogonal structuring elements of length 4. It is noted that the square structuring element has left significant amounts of 'pepper' noise in the image. Whilst the orthogonal structuring elements have been more successful in this respect, the distortion of the image is significant, most notably in the suppression of highlights such as those on the plastic rings.

The results for the area open-closing of the noisy image are also presented in terms of the increase in signal to noise ratio as a function of area in figure B.2. It may be expected that since impulsive noise affects isolated pixels, an area threshold of 1 should result in the maximum removal of both 'salt' and 'pepper' whilst disturbing image content the least. However, closer inspection of the noisy image (Figure B.1(a)) shows that the impulsive noise has in fact affected several image regions of more than one pixel in size. As such, the best increase according to our measure is achieved for an area threshold of 5 and yields a 19.9dB increase, more than 2dB greater than the median filter. Figure B.7 shows the result of performing an area open-closing with this area threshold.

Using the λ-max operator regional maxima with a grey level difference from their highest neighbouring region greater than λ are decreased by λ grey levels. The λ-min operator has the dual effect on regional minima. By performing the λ-max and λ-min operators in succession we have another open-closing filter capable of removing salt and pepper noise. The drawback of this filter is that regional extrema are suppressed by only λ grey levels and as such a large value of λ is required. This is notable in the plot of SNR increase as a function of λ presented in Figure B.2 in which the noise reduction has not reached a maximum before $\lambda = 10$. Figure B.8 shows the plot of noise reduction as a function of λ for values of λ up to 100. The maximum noise reduction is achieved for $\lambda = 64$ and provides an increase of 2.8dB, a value far below those achieved thus far. This value is due simply to the fact that



(a)

(b)

Figure B.6: The noisy image of 'girl' after morphological filtering with a square structuring element of size 2 (a), and orthogonal structuring elements of length 4 (b).

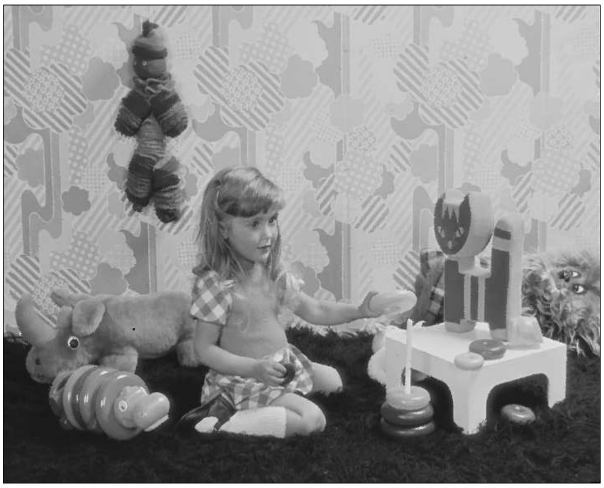


Figure B.7: The noisy image of 'girl' after application of an area open-closing with area threshold 5.

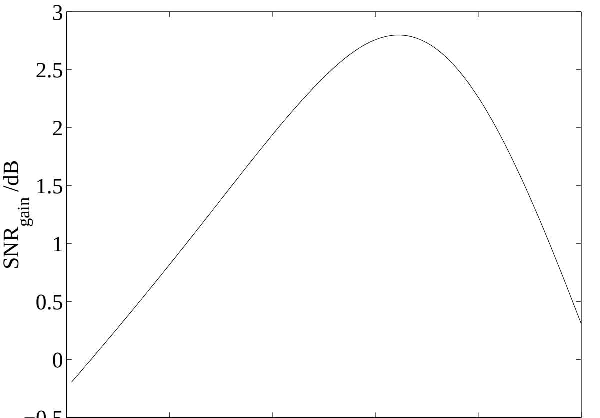


Figure B.8: The increase in signal to noise ratio $\mathrm{SNR}_{\mathrm{gain}}$ after application of the λ-max filter to the image of 'girl' corrupted with 10% impulsive noise as a function of λ.

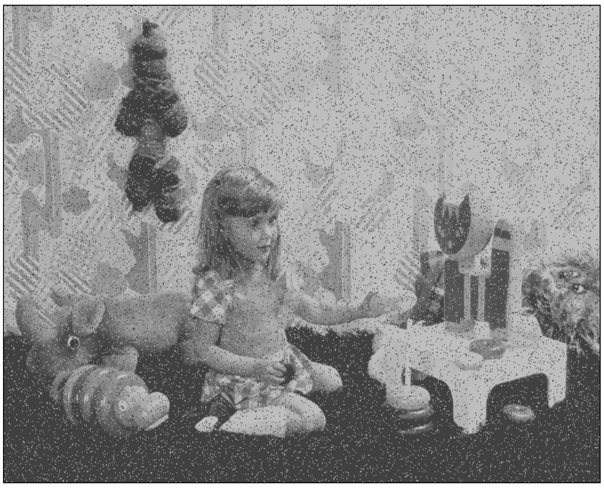


Figure B.9: The noisy image of 'girl' after filtering with the λ-max / λ-min filter with $\lambda = 64$.

a value of $\lambda = 64$ results in significant distortion of the image information, a fact which is clear from Figure B.9 in which the noise is still clearly visible.

The final strategy we shall employ involves the application of the adaptive λ-max and adaptive λ-min operators. In the case of the adaptive λ-max operator, regional maxima are suppressed not only by λ grey levels, but by their difference in grey level from their highest neighbour. The adaptive λ-min operator has the dual effect on the image's regional minima. The results of the application of the adaptive λ-max / λ-min operators are presented in Figure B.2 from which it is clear that the filter outperforms all other strategies giving an increase of 20.8dB. The noisy image of girl after filtering using this strategy is presented in Figure B.10.

The maximum attainable noise reduction for each noise reduction strategy are presented along with the parameter used to achieve it in Table B.1

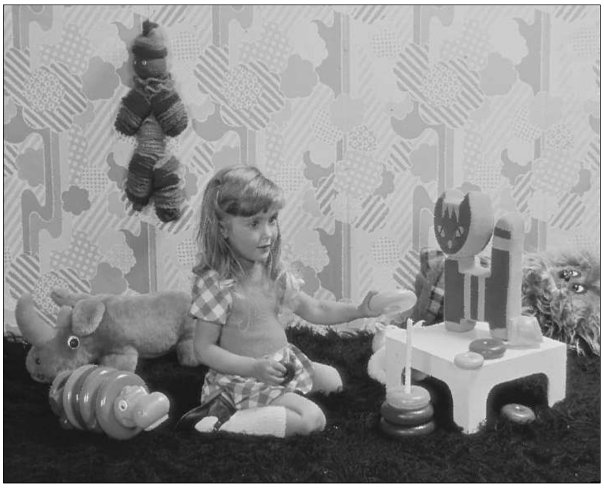


Figure B.10: The noisy image of ‘girl’ after filtering with the adaptive λ-max / λ-min filter with $\lambda = 1$.

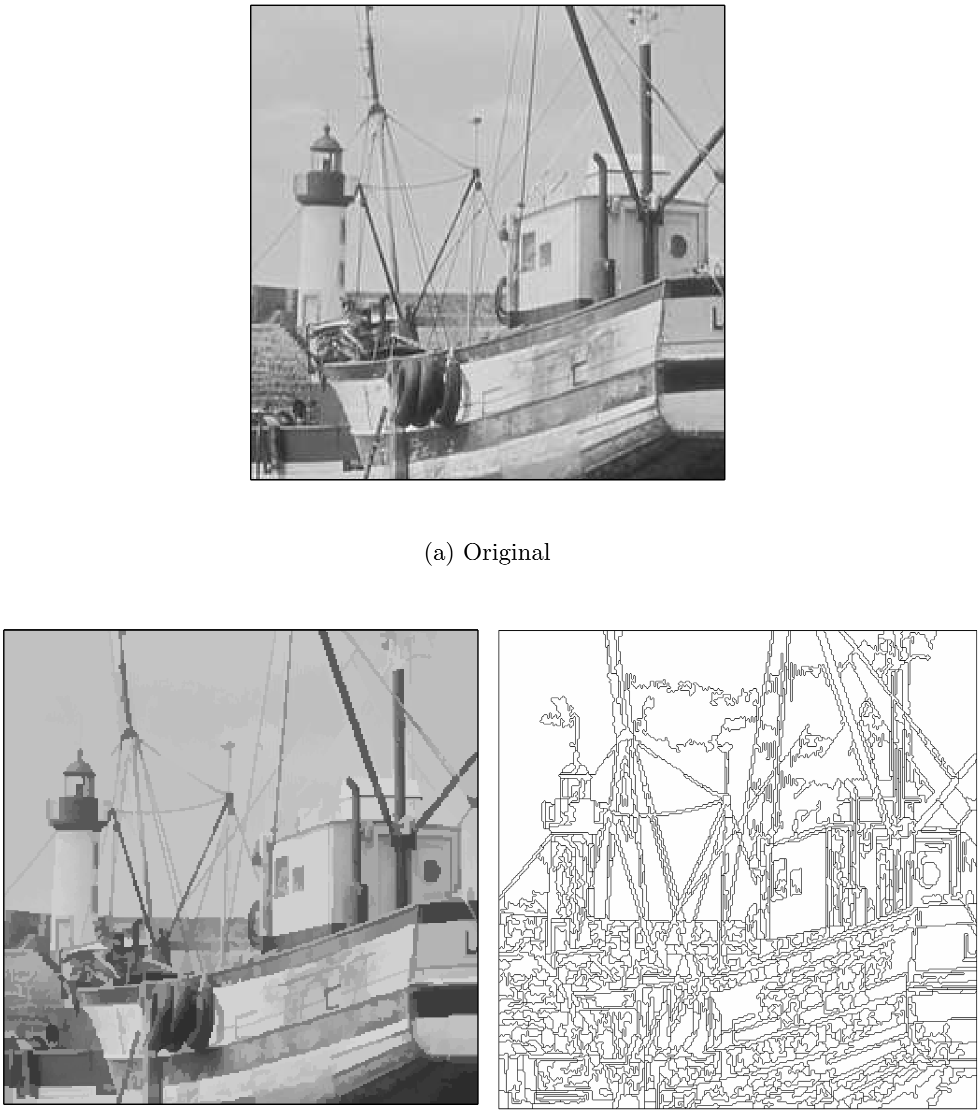
Method	Parameter	$\mathrm{SNR}_{\mathrm{gain}}$
Median Filter	3×3	17.9dB
Morphological Square SE	2×2	12.0dB
Morphological Orth. Line SEs	1×4 / 4×1	12.8dB
Area open-closing	$a = 5$	19.9dB
λ-max / λ-min	$\lambda = 64$	2.8dB
Adaptive λ-max / λ-min	$\lambda = 1$	20.8dB

Table B.1: A table of maximum noise reduction attainable for various strategies acting upon the ‘noisy’ image of ‘girl’ presented in Figure B.1(a).

Appendix C

Additional Results of the Adaptive λ-max and Grey Scale Merging Operators

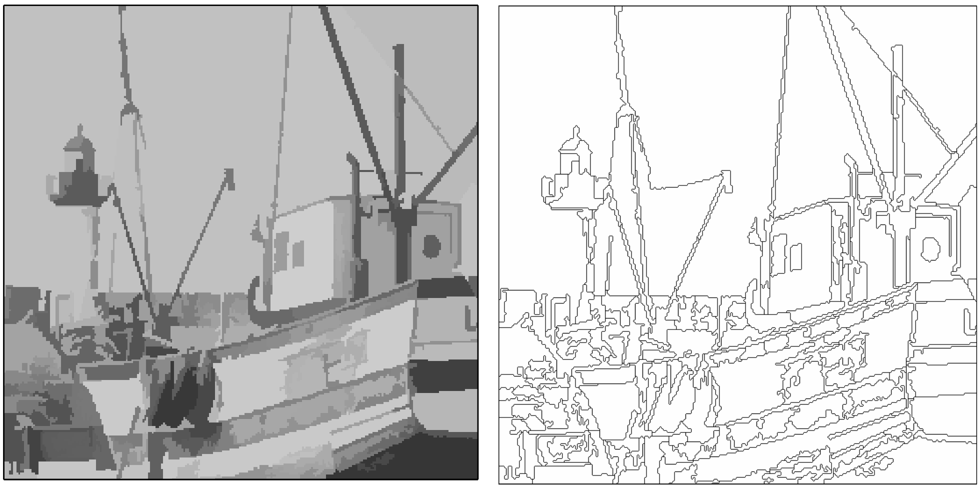
This appendix contains further examples of segmentations achieved using the adaptive λ-max operator followed by the grey scale-merging operator as presented in Section 4.3. The results are chosen to correspond with the minima of the weighted total coding cost surfaces for the stated value of α. The result of the segmentation of the image 'boat' with $\alpha = 0.75$ is omitted since is describes a single region (see Table 4.1. All the original images are 256 × 256 pixels and though the resulting outline images are necessarily larger, they have been scaled accordingly.



(a) Original

(b) $\alpha = 0.5$ Mean replaced.

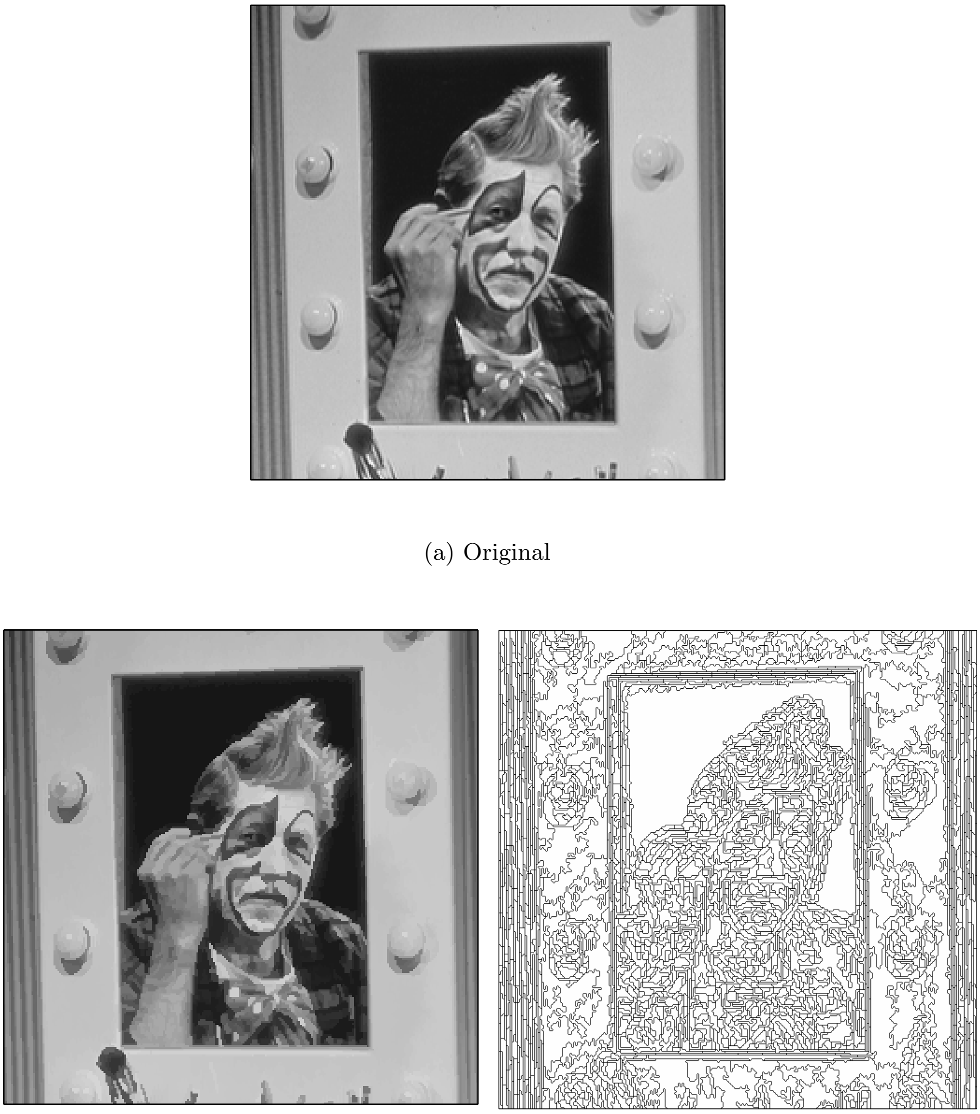
(c) $\alpha = 0.5$ Outline.



(d) $\alpha = 0.6$ Mean replaced.

(e) $\alpha = 0.6$ Outline.

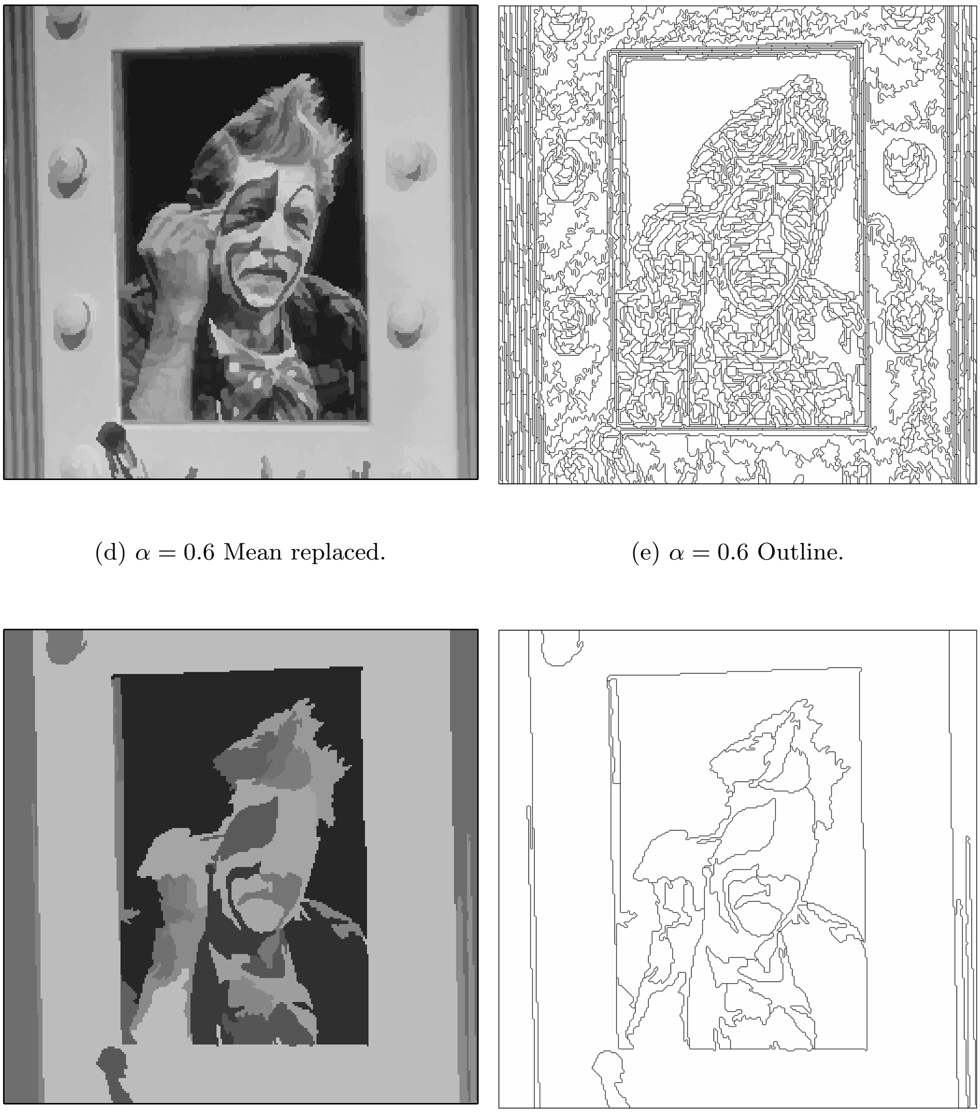
Figure C.1: The original image of 'boat' (a) and the mean replaced and outline images for $\alpha = 0.5,\ 0.6$ and 0.75 (b) to (g).



(a) Original

(b) $\alpha = 0.5$ Mean replaced.

(c) $\alpha = 0.5$ Outline.



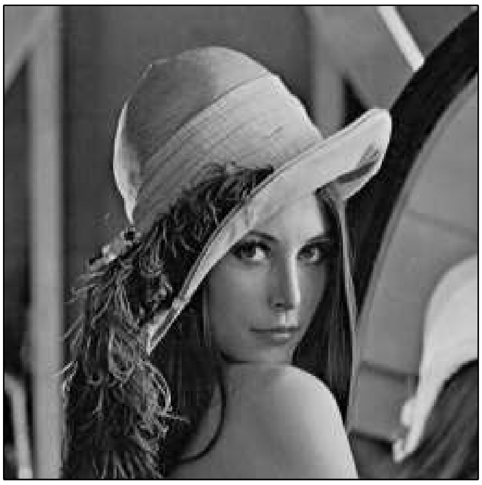
(d) $\alpha = 0.6$ Mean replaced.

(e) $\alpha = 0.6$ Outline.

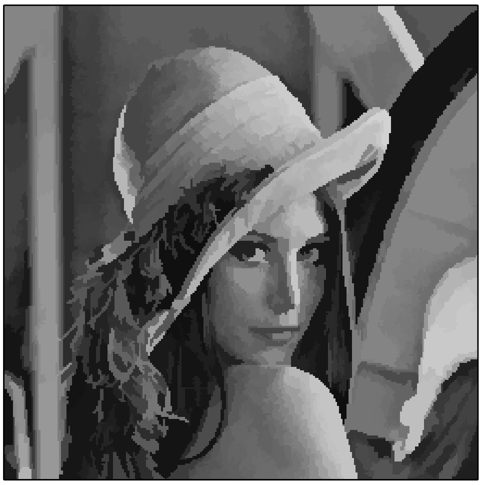
(f) $\alpha = 0.75$ Mean replaced.

(g) $\alpha = 0.75$ Outline.

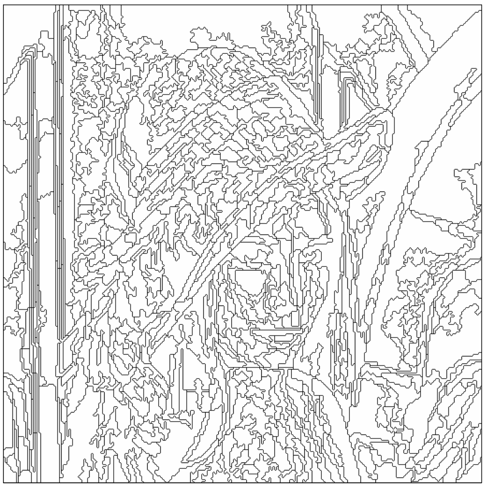
Figure C.2: The original image of 'clown' (a) and the mean replaced and outline images for $\alpha = 0.5,\ 0.6$ and 0.75 (b) to (g).



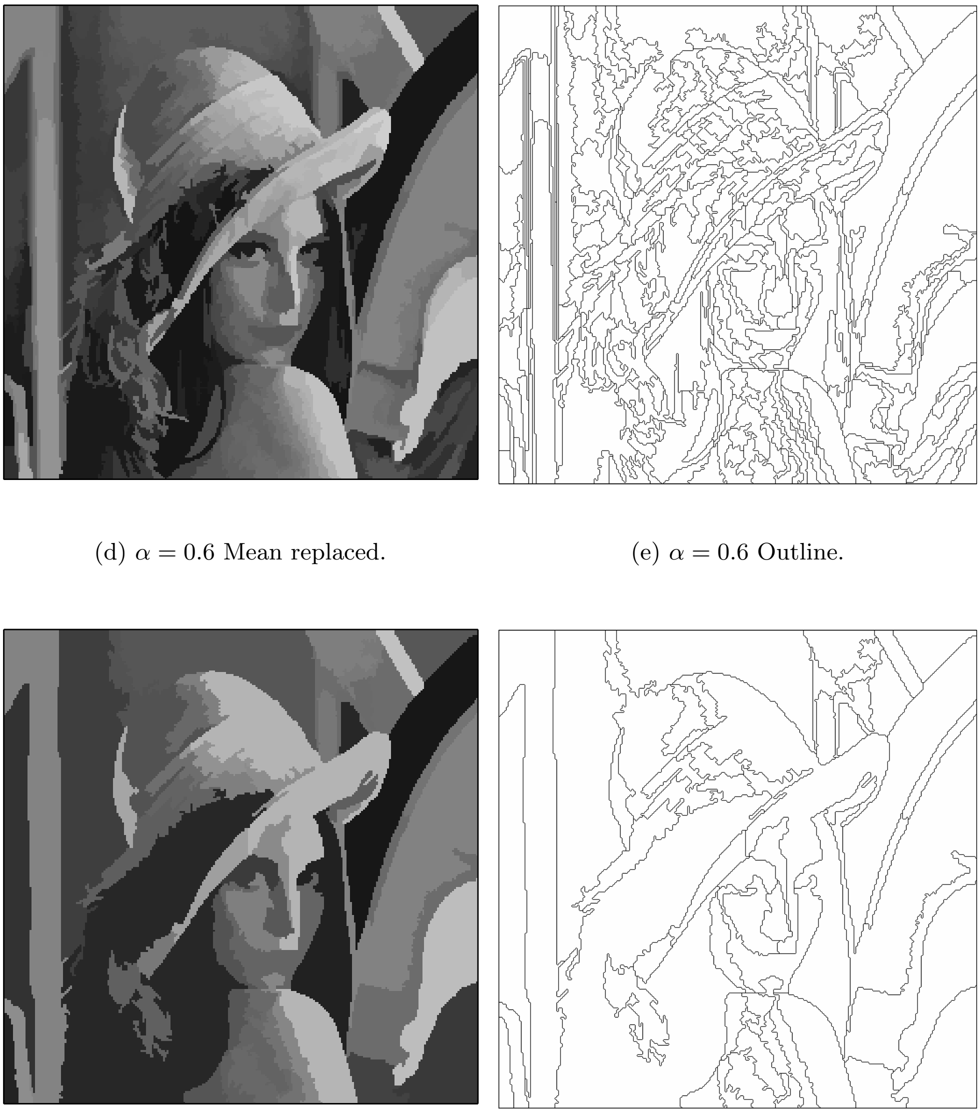
(a) Original



(b) $\alpha = 0.5$ Mean replaced.



(c) $\alpha = 0.5$ Outline.



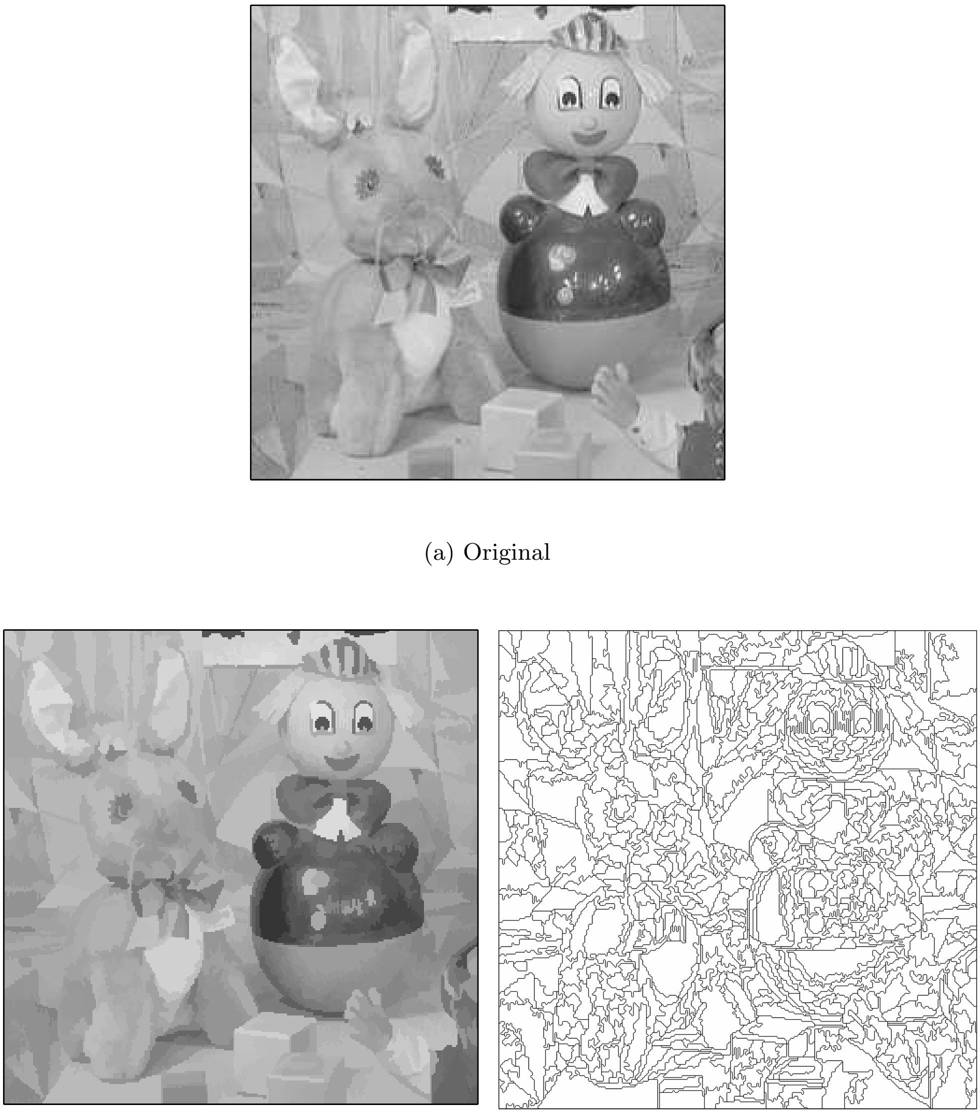
(d) $\alpha = 0.6$ Mean replaced.

(e) $\alpha = 0.6$ Outline.

(f) $\alpha = 0.75$ Mean replaced.

(g) $\alpha = 0.75$ Outline.

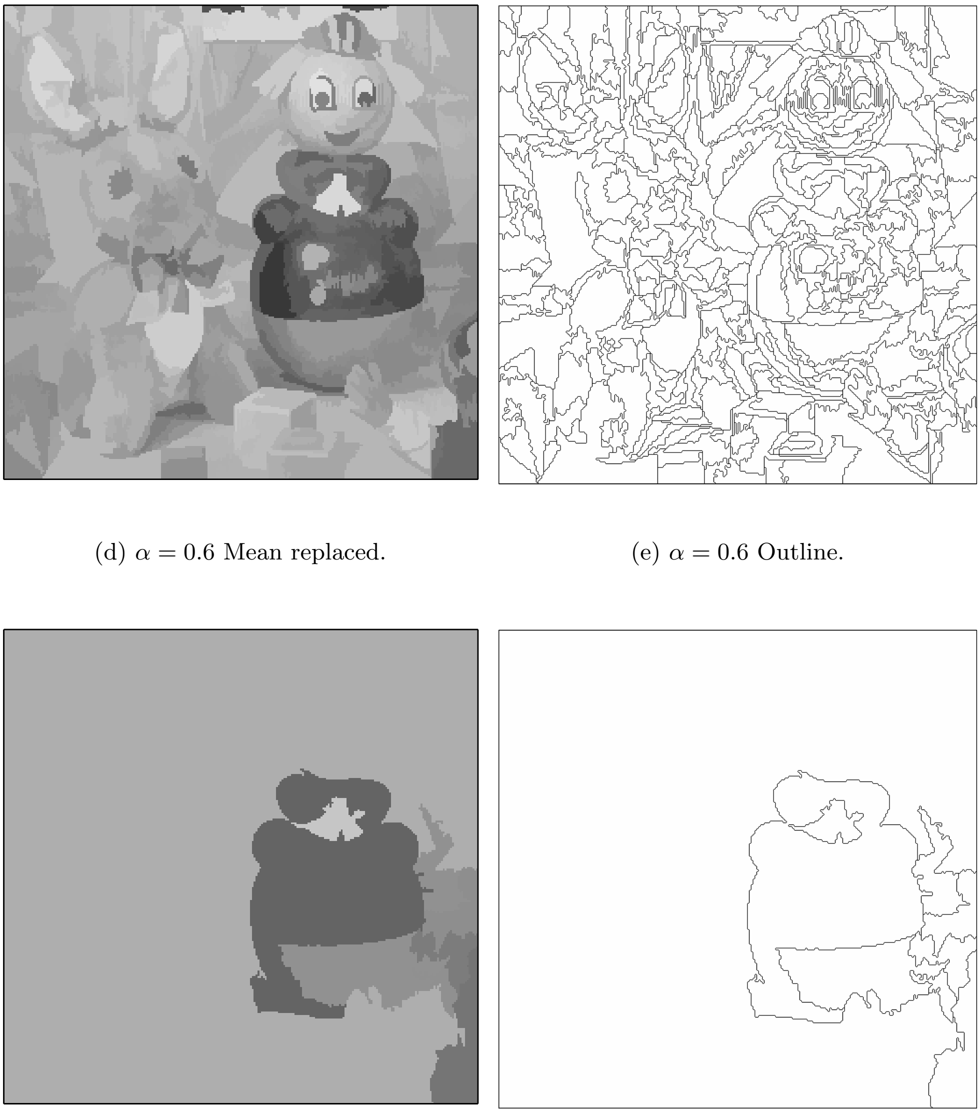
Figure C.3: The original image of ‘lenna’ (a) and the mean replaced and outline images for $\alpha = 0.5,\ 0.6$ and 0.75 (b) to (g).



(a) Original

(b) $\alpha = 0.5$ Mean replaced.

(c) $\alpha = 0.5$ Outline.



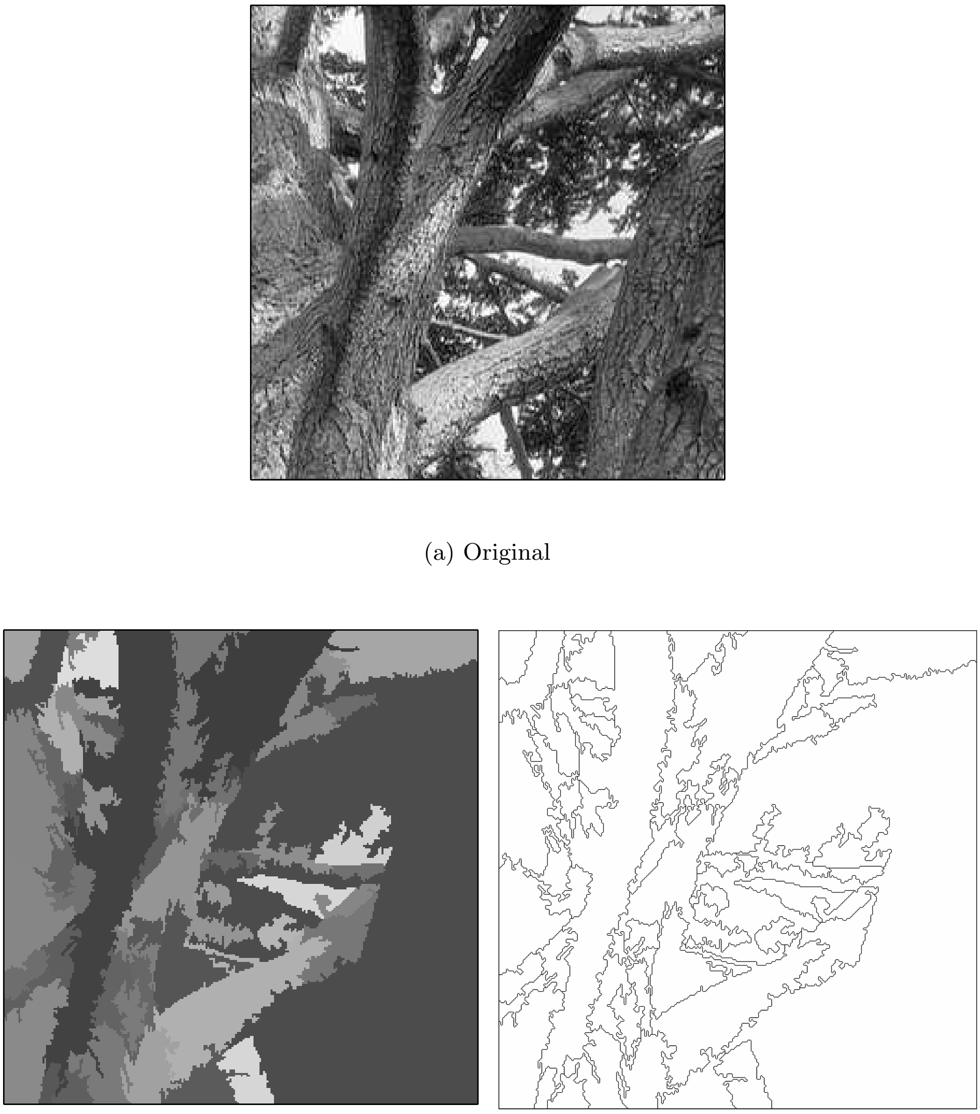
(d) $\alpha = 0.6$ Mean replaced.

(e) $\alpha = 0.6$ Outline.

(f) $\alpha = 0.75$ Mean replaced.

(g) $\alpha = 0.75$ Outline.

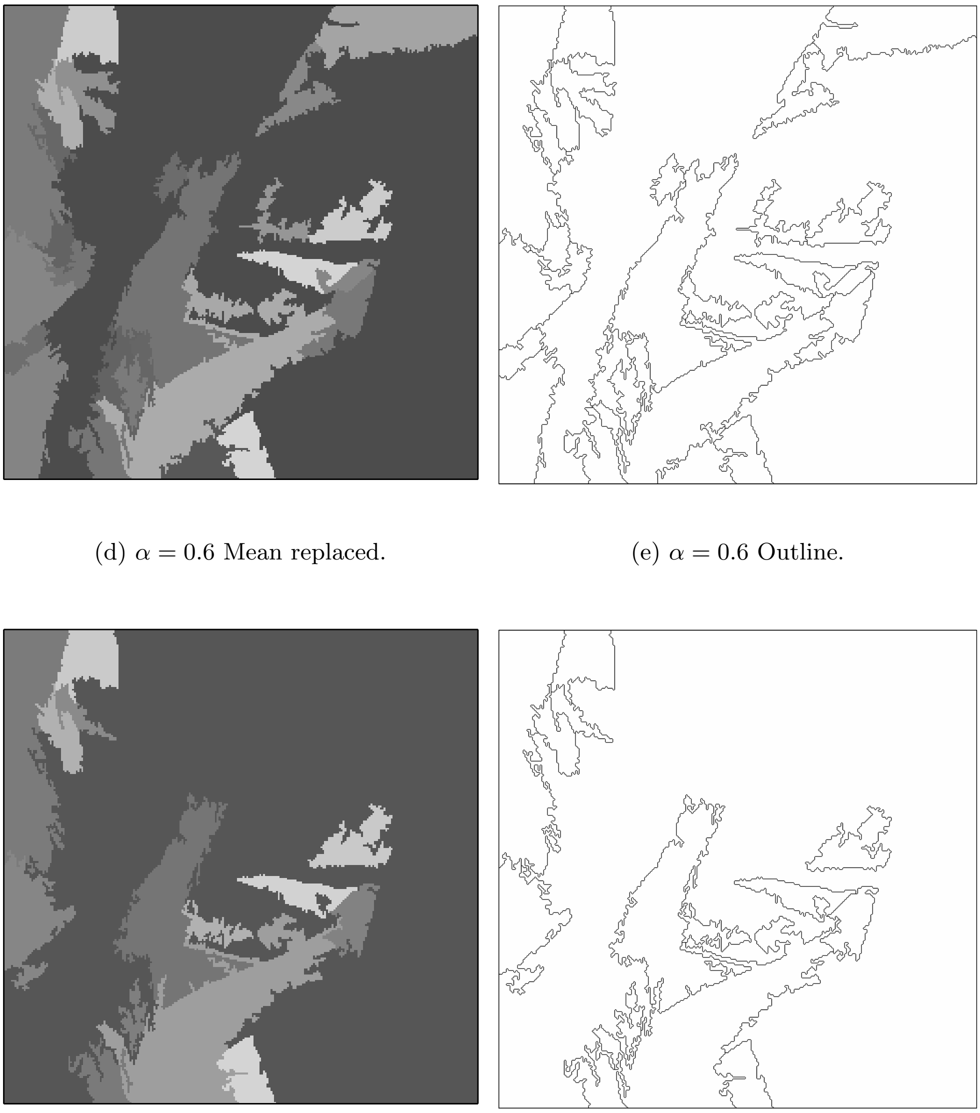
Figure C.4: The original image of ‘toys’ (a) and the mean replaced and outline images for $\alpha = 0.5,\ 0.6$ and 0.75 (b) to (g).



(a) Original

(b) $\alpha = 0.5$ Mean replaced.

(c) $\alpha = 0.5$ Outline.



(d) $\alpha = 0.6$ Mean replaced.

(e) $\alpha = 0.6$ Outline.

(f) $\alpha = 0.75$ Mean replaced.

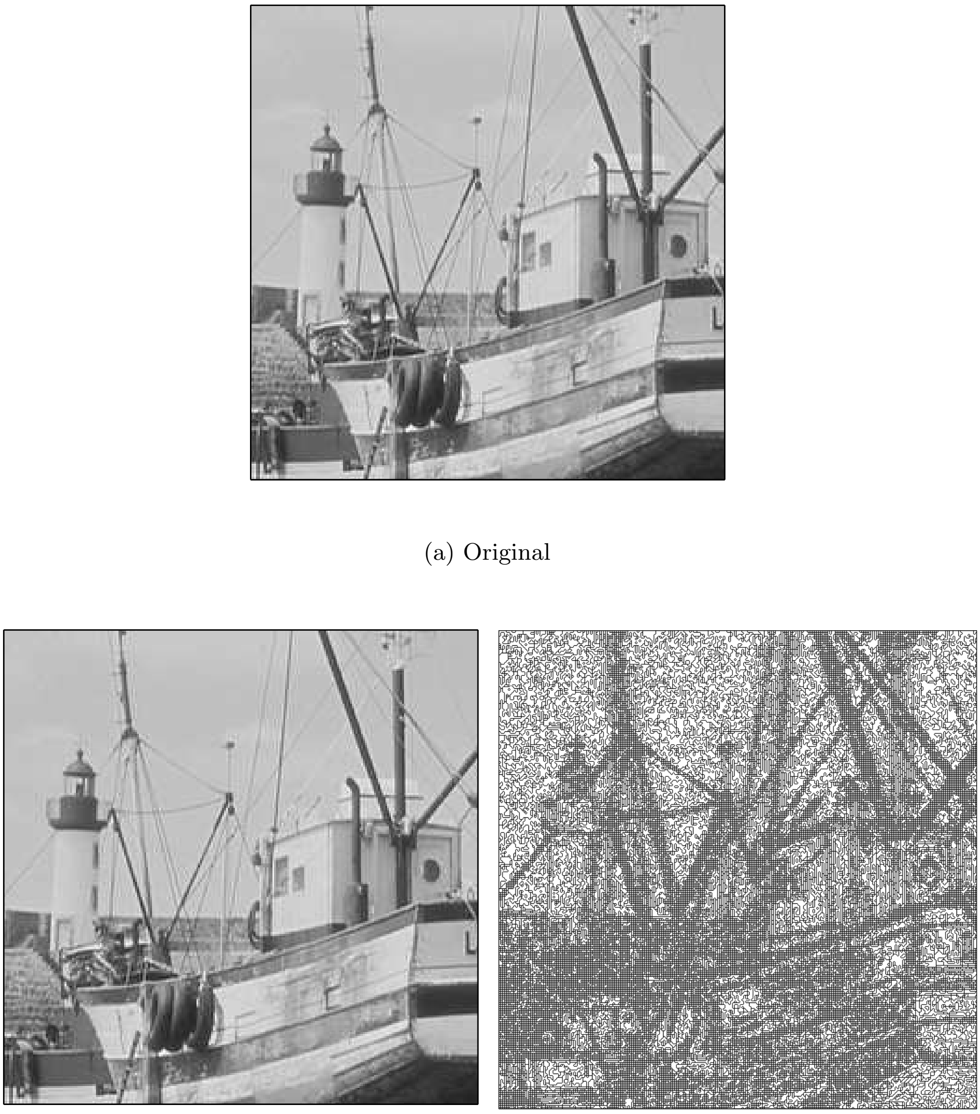
(g) $\alpha = 0.75$ Outline.

Figure C.5: The original image of 'tree' (a) and the mean replaced and outline images for $\alpha = 0.5,\ 0.6$ and 0.75 (b) to (g).

Appendix D

Additional Results of the Entropy Minimising Operator

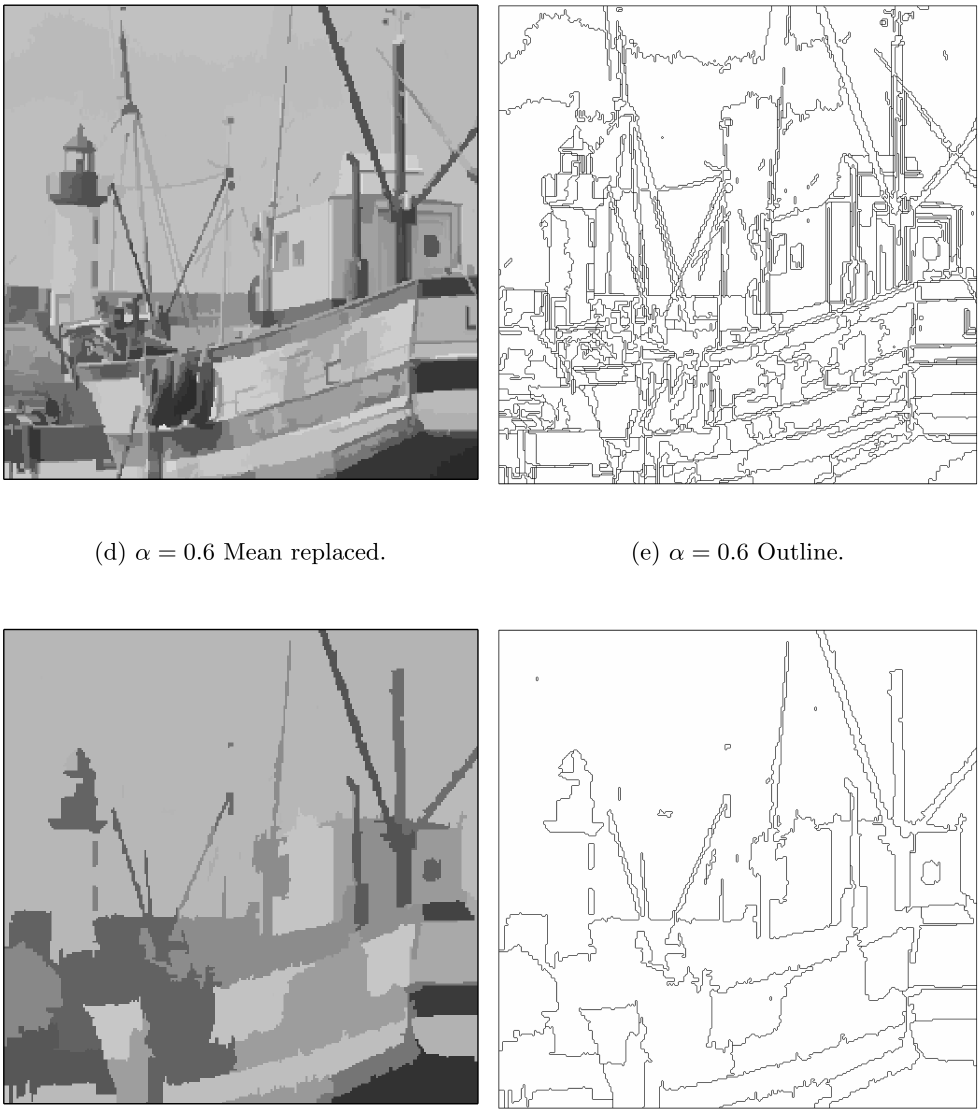
This appendix contains further examples of segmentations achieved using the entropy minimising operator as presented in Section 4.3. The results presented correspond to values of $\alpha = 0.5,\ 0.6$ and 0.75 though results for the image of 'tree' are omitted since they are as the original for $\alpha = 0.5$ and a single region for $\alpha = 0.75$ (see Table 4.2). All the original images are 256×256 pixels and though the resulting outline images are necessarily larger, they have been scaled accordingly.



(a) Original

(b) $\alpha = 0.5$ Mean replaced.

(c) $\alpha = 0.5$ Outline.



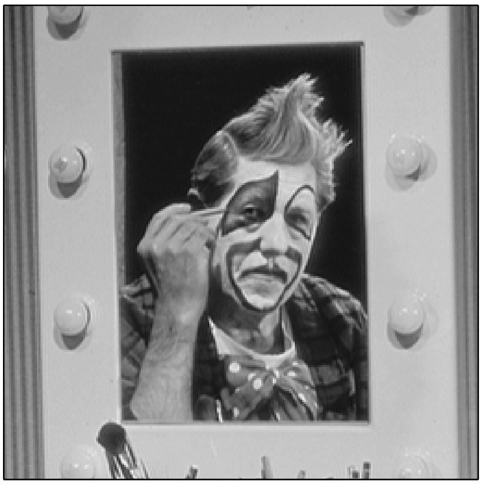
(d) $\alpha = 0.6$ Mean replaced.

(e) $\alpha = 0.6$ Outline.

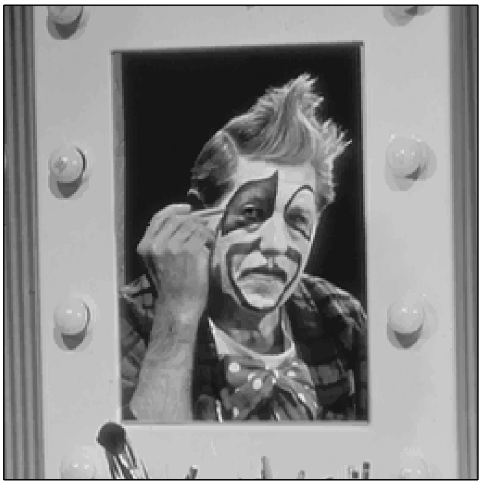
(f) $\alpha = 0.75$ Mean replaced.

(g) $\alpha = 0.75$ Outline.

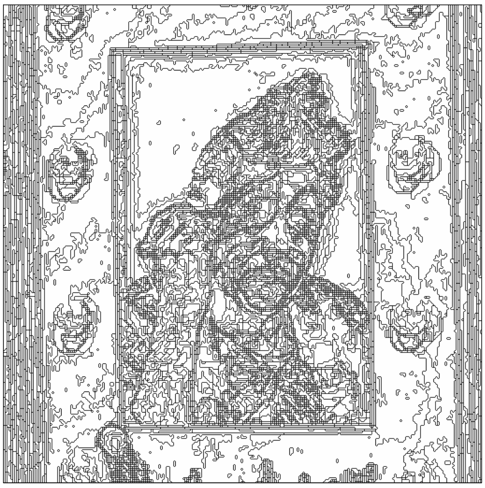
Figure D.1: The original image of 'boat' (a) and the mean replaced and outline images for $\alpha = 0.5,\ 0.6$ and 0.75 (b) to (g).



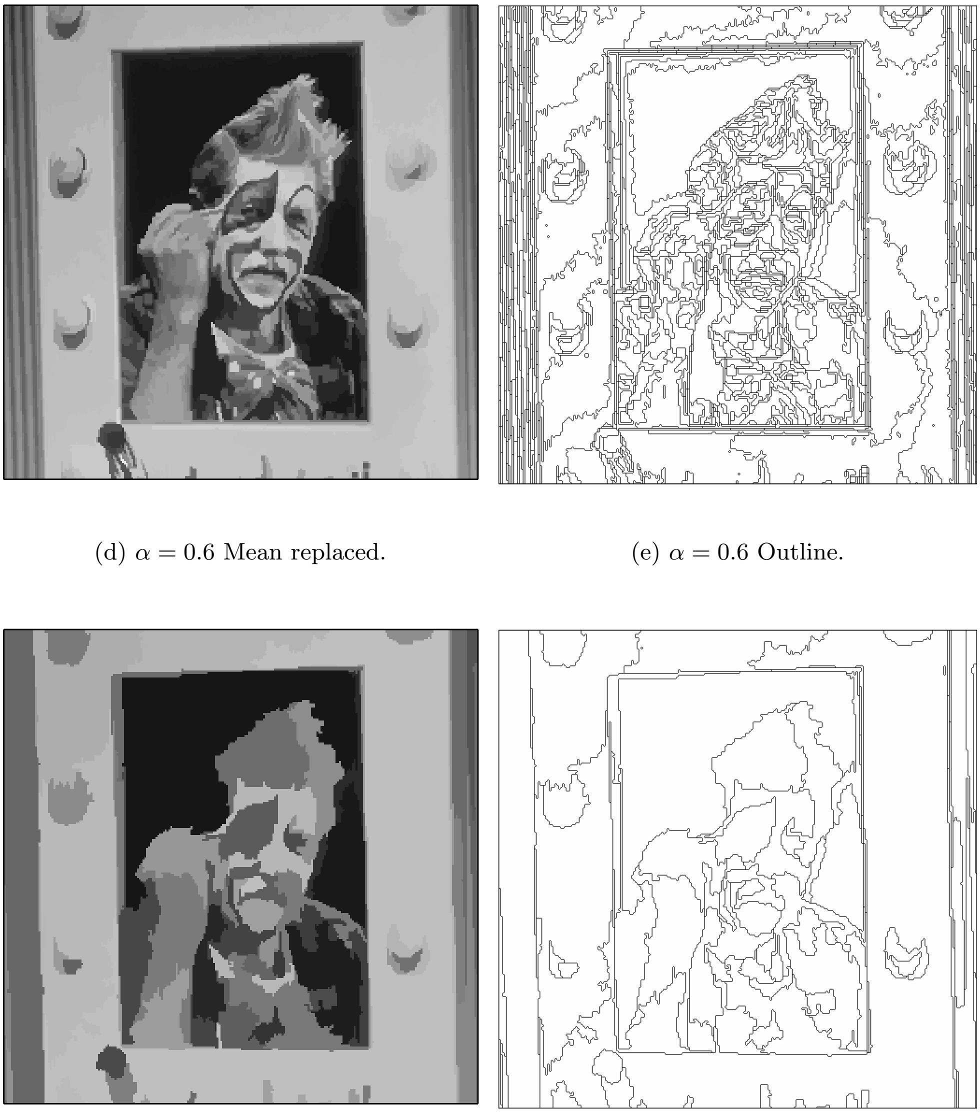
(a) Original



(b) $\alpha = 0.5$ Mean replaced.



(c) $\alpha = 0.5$ Outline.



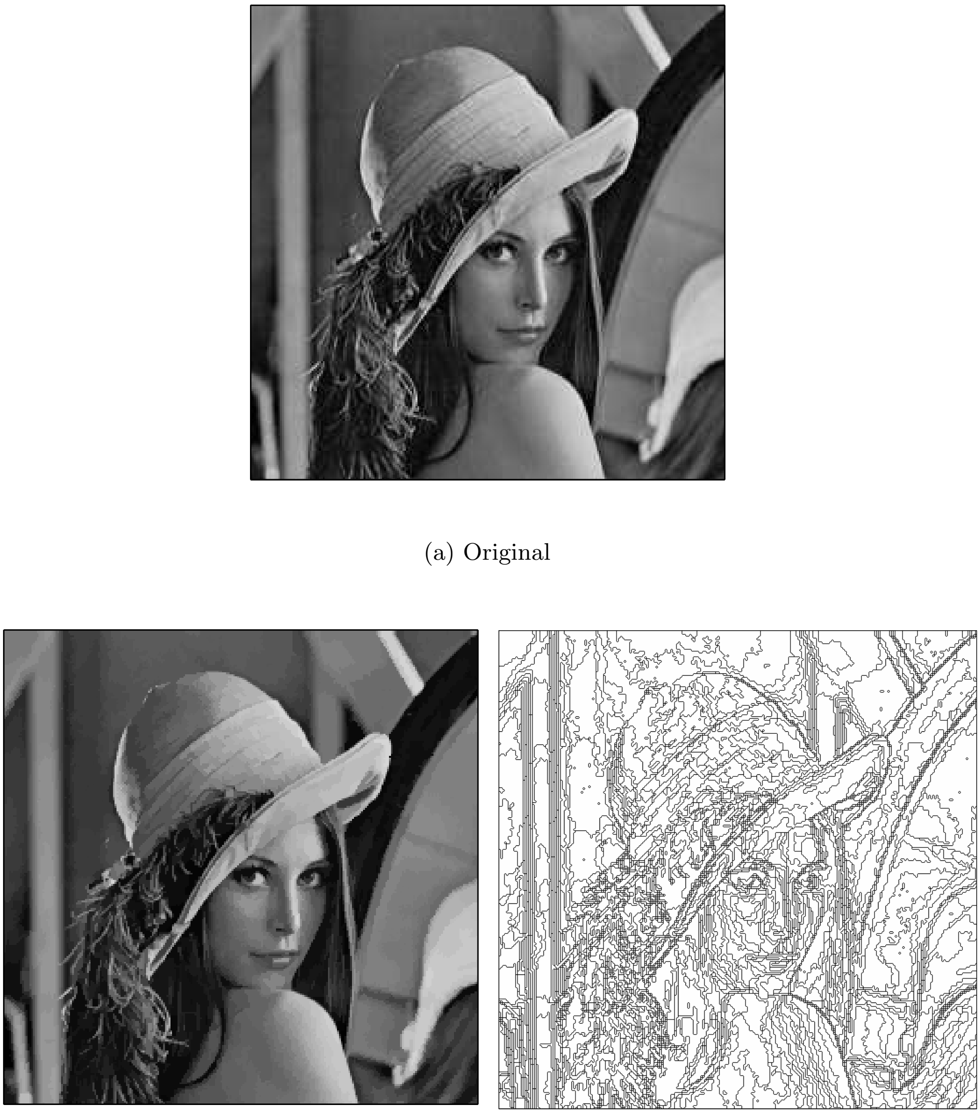
(d) $\alpha = 0.6$ Mean replaced.

(e) $\alpha = 0.6$ Outline.

(f) $\alpha = 0.75$ Mean replaced.

(g) $\alpha = 0.75$ Outline.

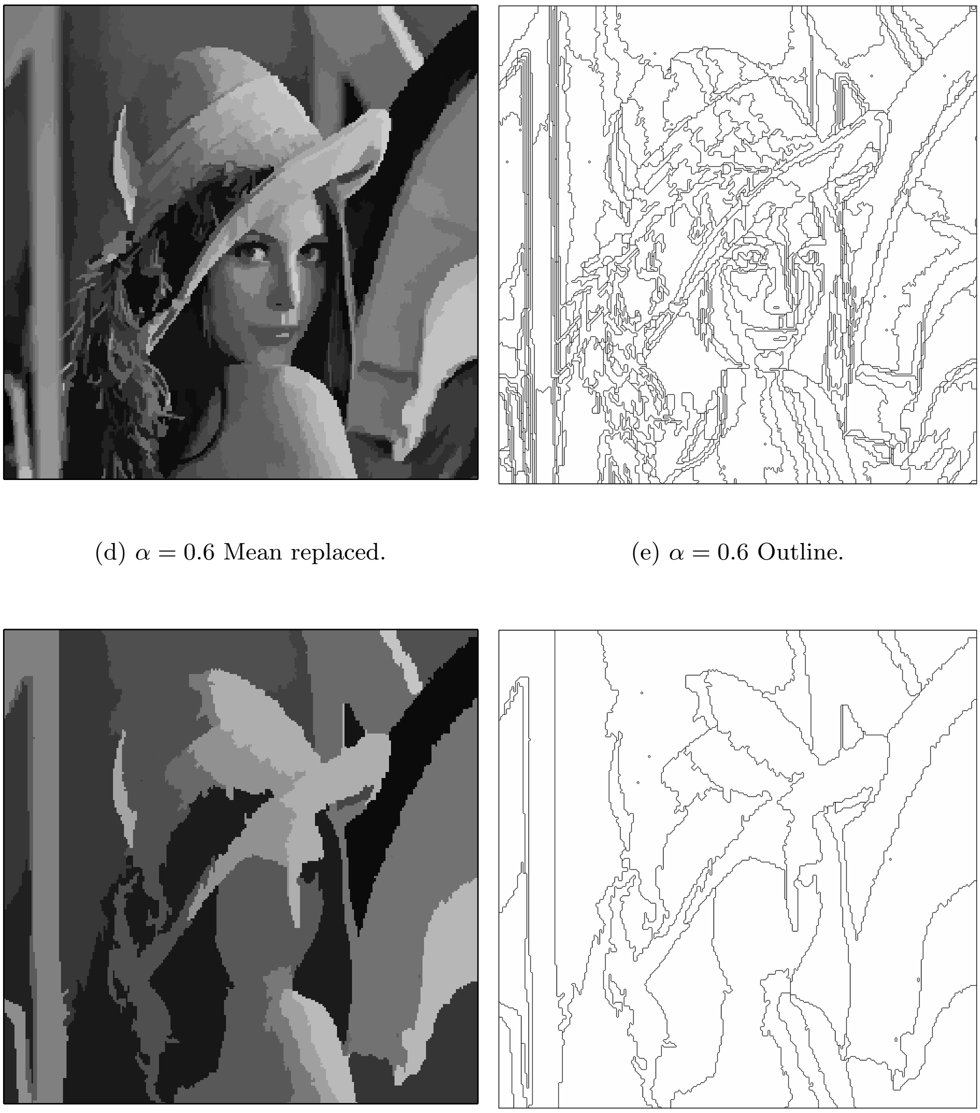
Figure D.2: The original image of 'clown' (a) and the mean replaced and outline images for $\alpha = 0.5,\ 0.6$ and 0.75 (b) to (g).



(a) Original

(b) $\alpha = 0.5$ Mean replaced.

(c) $\alpha = 0.5$ Outline.



(d) $\alpha = 0.6$ Mean replaced.

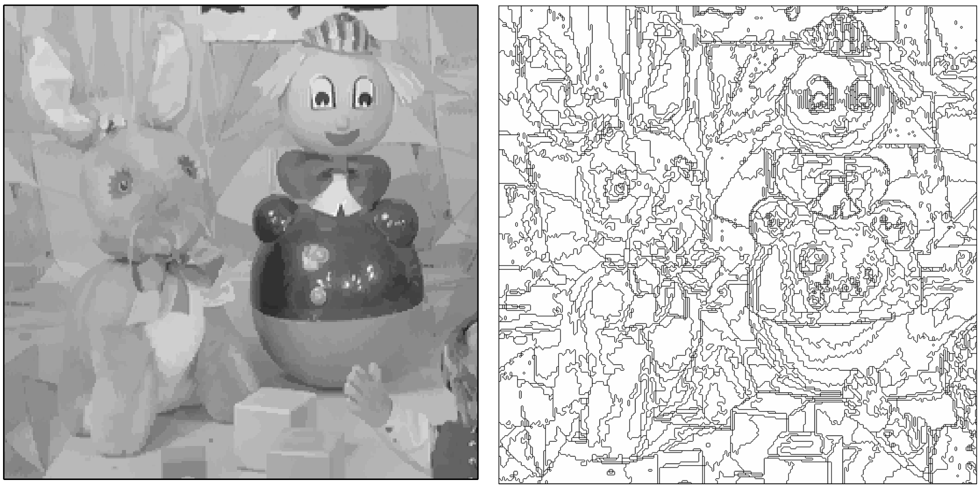
(e) $\alpha = 0.6$ Outline.

(f) $\alpha = 0.75$ Mean replaced.

(g) $\alpha = 0.75$ Outline.

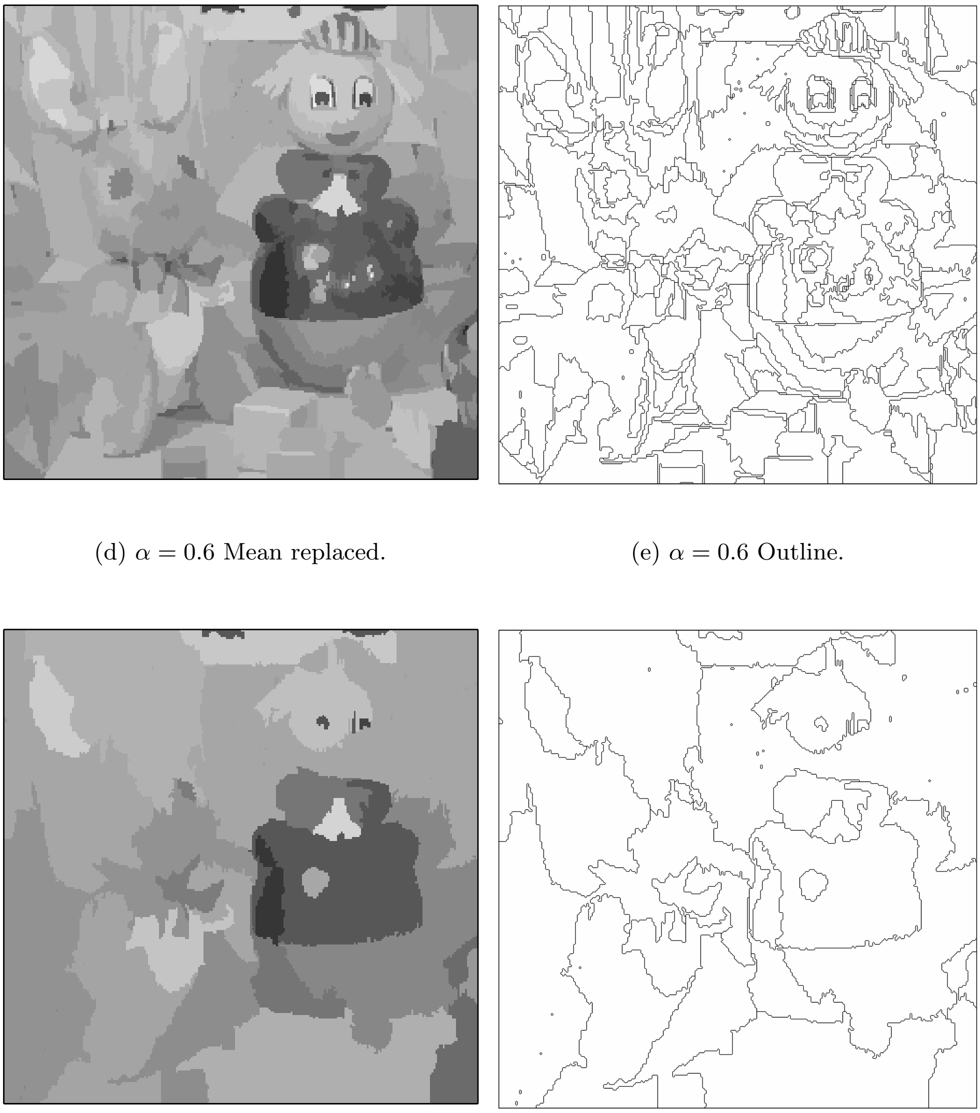
Figure D.3: The original image of 'lenna' (a) and the mean replaced and outline images for $\alpha = 0.5,\ 0.6$ and 0.75 (b) to (g).

(a) Original



(b) $\alpha = 0.5$ Mean replaced.

(c) $\alpha = 0.5$ Outline.



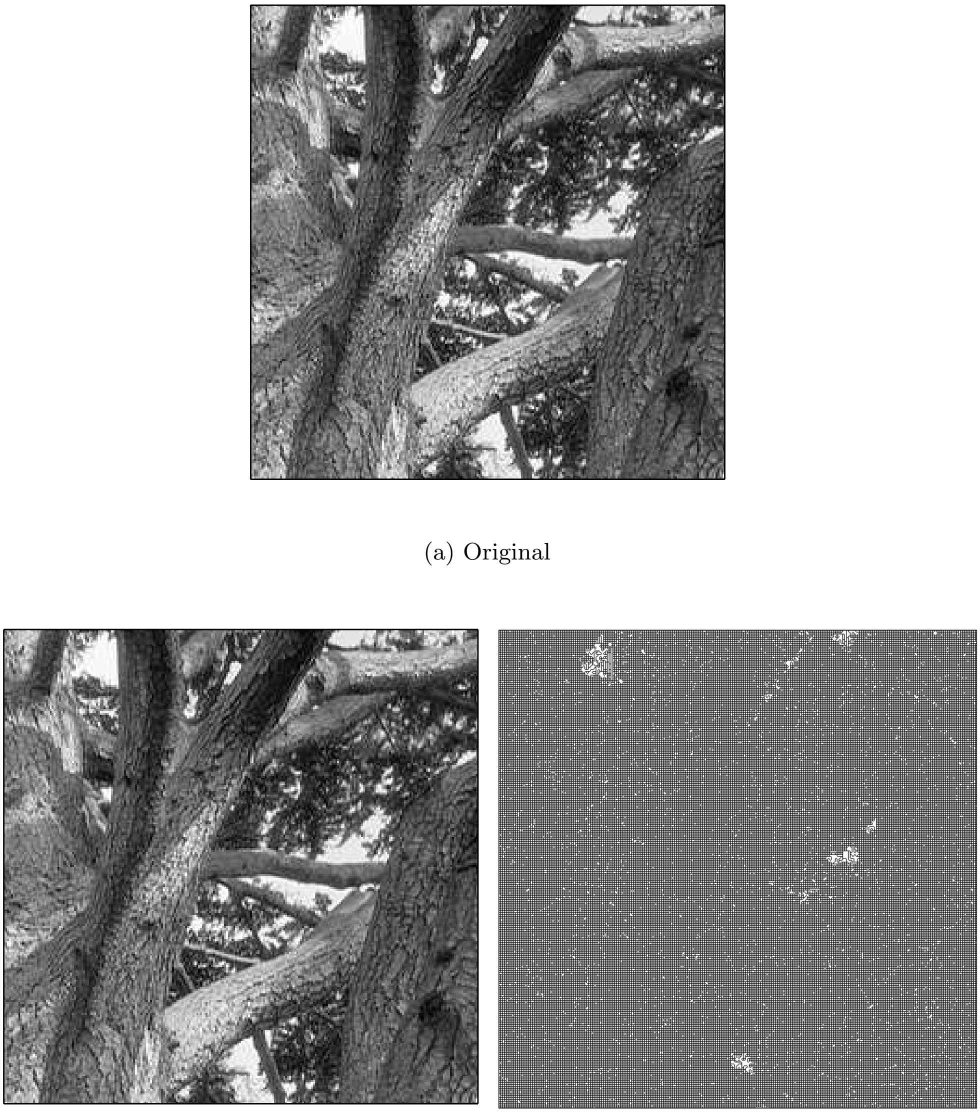
(d) $\alpha = 0.6$ Mean replaced.

(e) $\alpha = 0.6$ Outline.

(f) $\alpha = 0.75$ Mean replaced.

(g) $\alpha = 0.75$ Outline.

Figure D.4: The original image of 'toys' (a) and the mean replaced and outline images for $\alpha = 0.5,\ 0.6$ and 0.75 (b) to (g).



(a) Original

(b) $\alpha = 0.6$ Mean replaced.

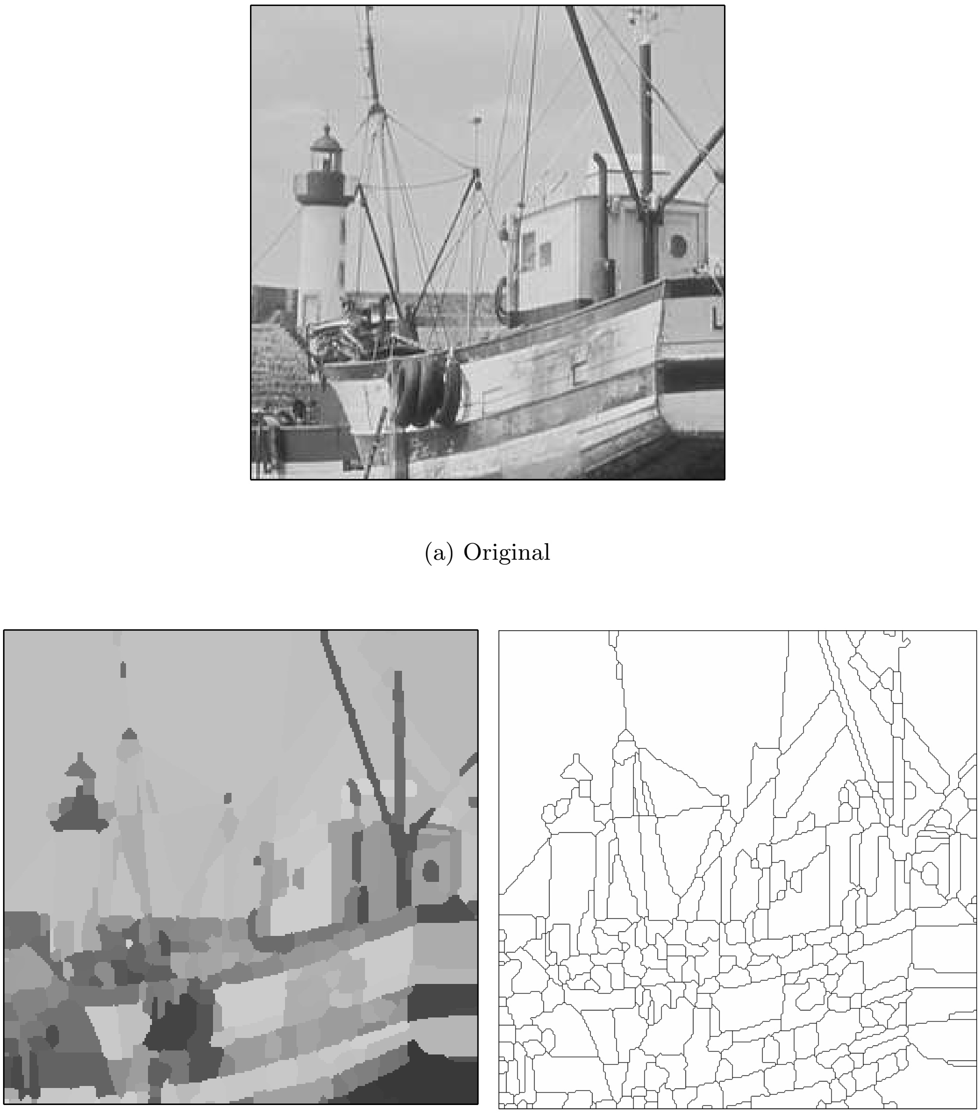
(c) $\alpha = 0.6$ Outline.

Figure D.5: The original image of ‘tree’ (a) and the mean replaced and outline images for $\alpha = 0.5$ (b) and (c).

Appendix E

Additional results for the Gradient Smoothing Watershed Segmentation Scheme

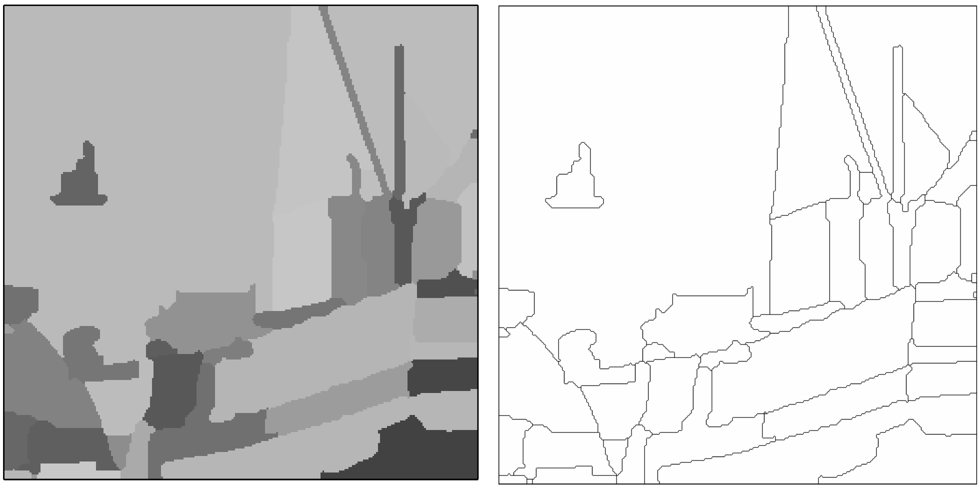
This appendix presents further examples of the segmentations produced using the gradient smoothing and watershed transform segmentation scheme presented in Section 5.2 for values of $\alpha = 0.5,\ 0.6$ and 0.75. Images for the segmentation of the image of both 'boat' and 'toys' using $\alpha = 0.6$ have been omitted as the results are identical to those produced using $\alpha = 0.5$. Images for the segmentations of the image of 'tree' for $\alpha = 0.6$ and 0.75 are omitted as they result in a single region. All the original images are 256×256 pixels and though the resulting outline images are necessarily larger, they have been scaled accordingly.



(a) Original

(b) $\alpha = 0.5$ Mean replaced.

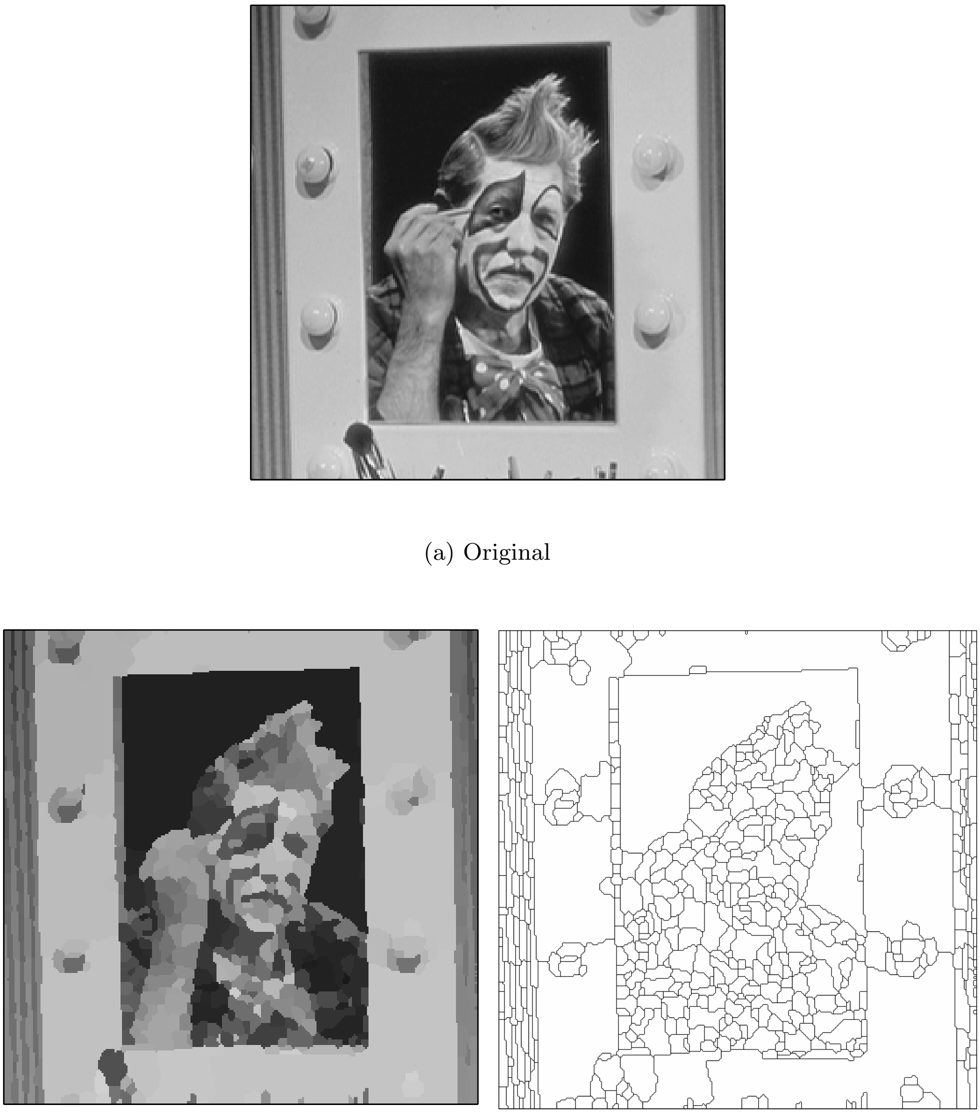
(c) $\alpha = 0.5$ Outline.



(d) $\alpha = 0.75$ Mean replaced.

(e) $\alpha = 0.75$ Outline.

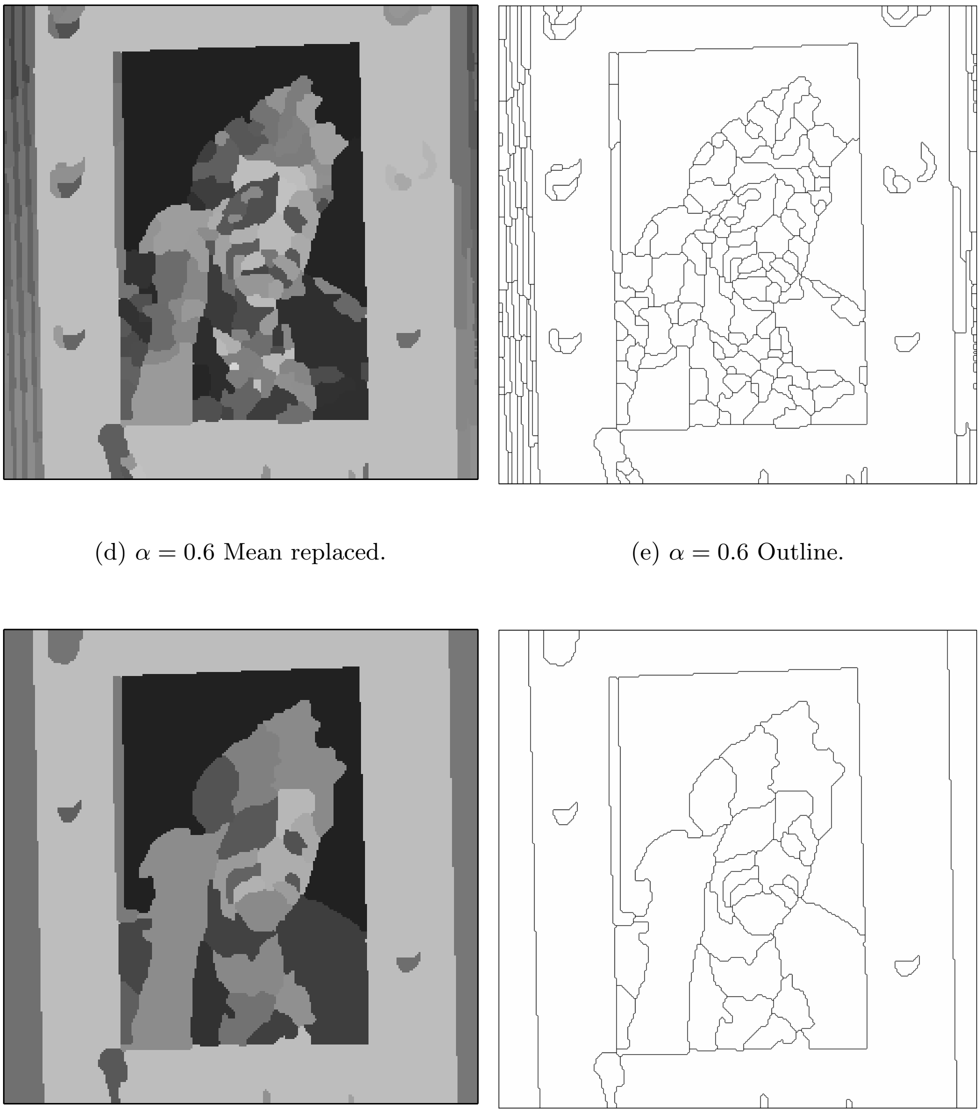
Figure E.1: The original image of 'boat' (a) and the mean replaced and outline images for $\alpha = 0.5$ and 0.75 (b) to (e).



(a) Original

(b) $\alpha = 0.5$ Mean replaced.

(c) $\alpha = 0.5$ Outline.



(d) $\alpha = 0.6$ Mean replaced.

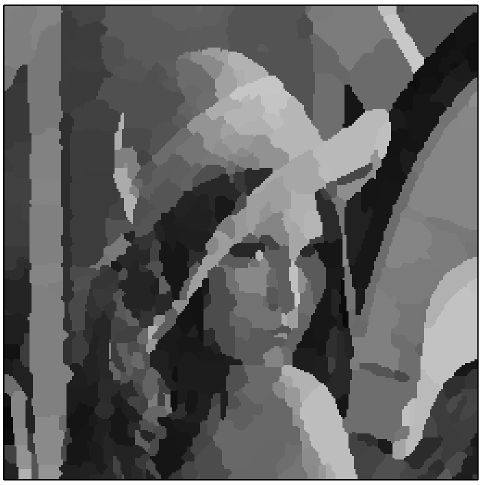
(e) $\alpha = 0.6$ Outline.

(f) $\alpha = 0.75$ Mean replaced.

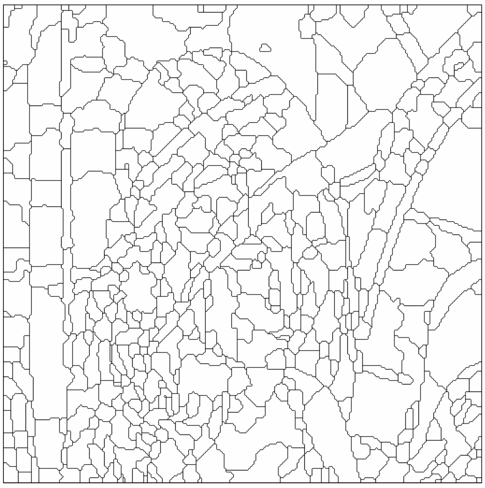
(g) $\alpha = 0.75$ Outline.

Figure E.2: The original image of ‘clown’ (a) and the mean replaced and outline images for $\alpha = 0.5,\ 0.6$ and 0.75 (b) to (g).

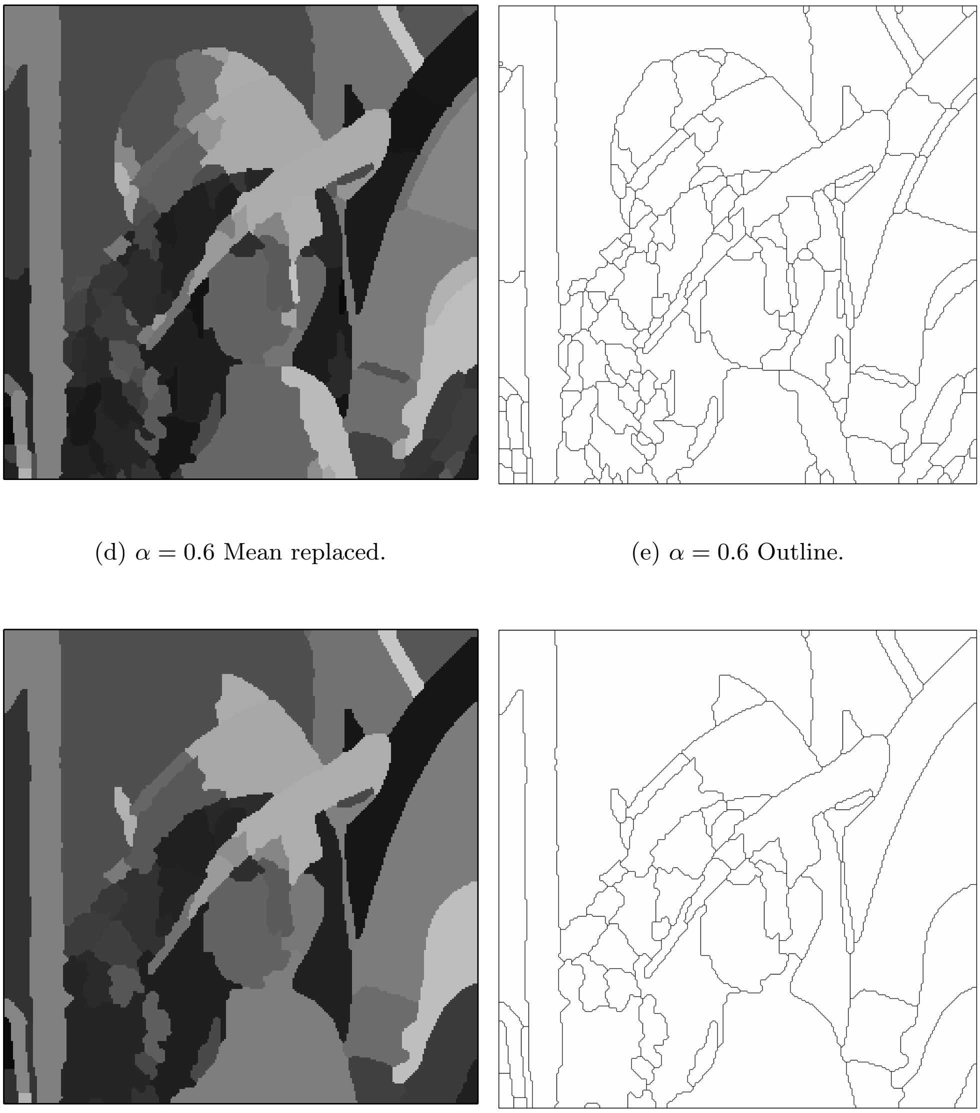
(a) Original



(b) $\alpha = 0.5$ Mean replaced.



(c) $\alpha = 0.5$ Outline.



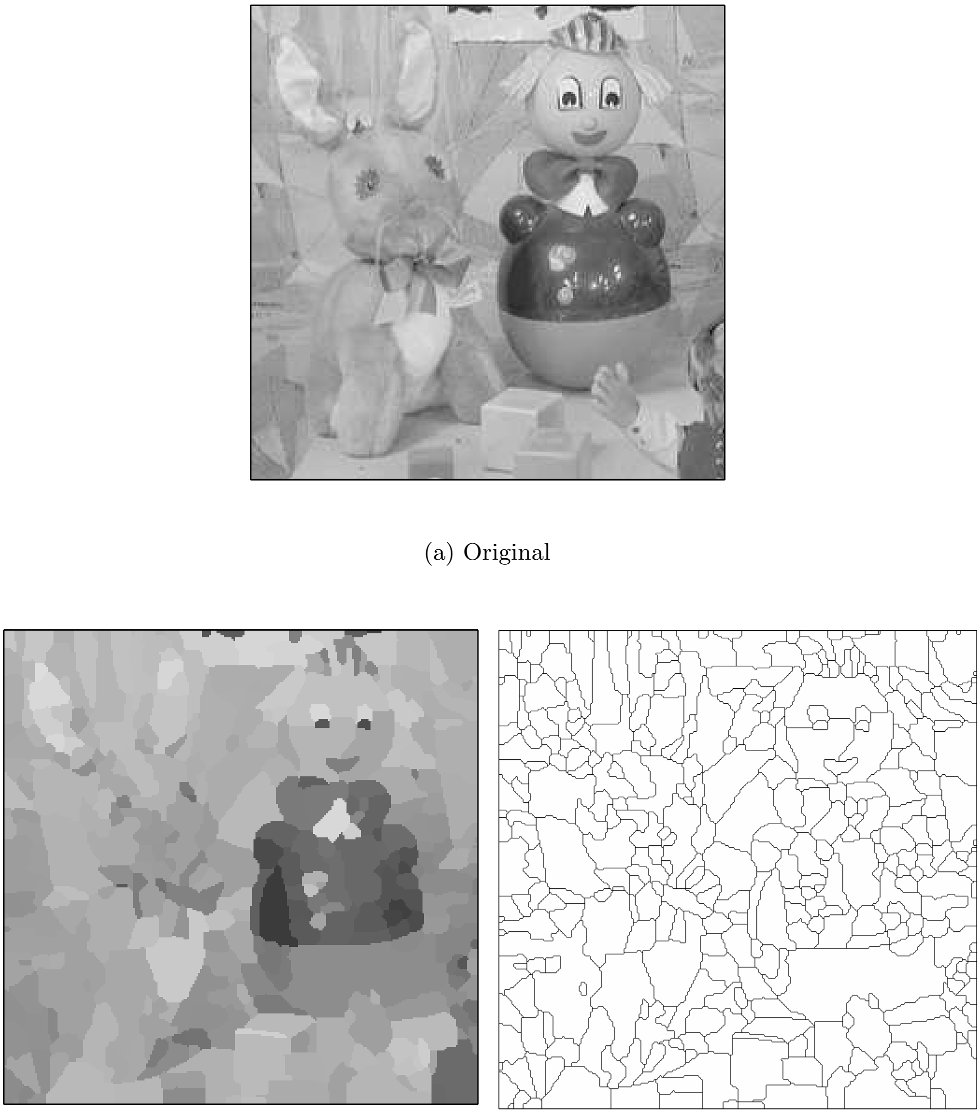
(d) $\alpha = 0.6$ Mean replaced.

(e) $\alpha = 0.6$ Outline.

(f) $\alpha = 0.75$ Mean replaced.

(g) $\alpha = 0.75$ Outline.

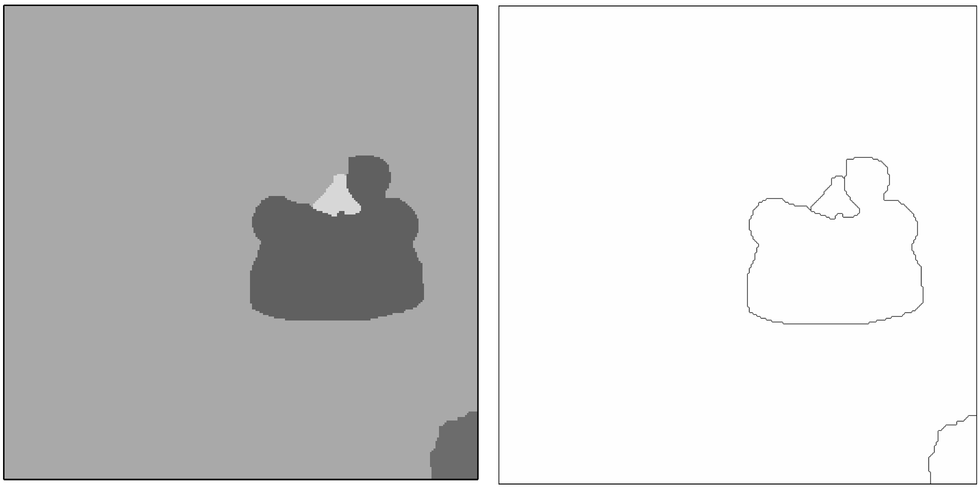
Figure E.3: The original image of 'lenna' (a) and the mean replaced and outline images for $\alpha = 0.5,\ 0.6$ and 0.75 (b) to (g).



(a) Original

(b) $\alpha = 0.5$ Mean replaced.

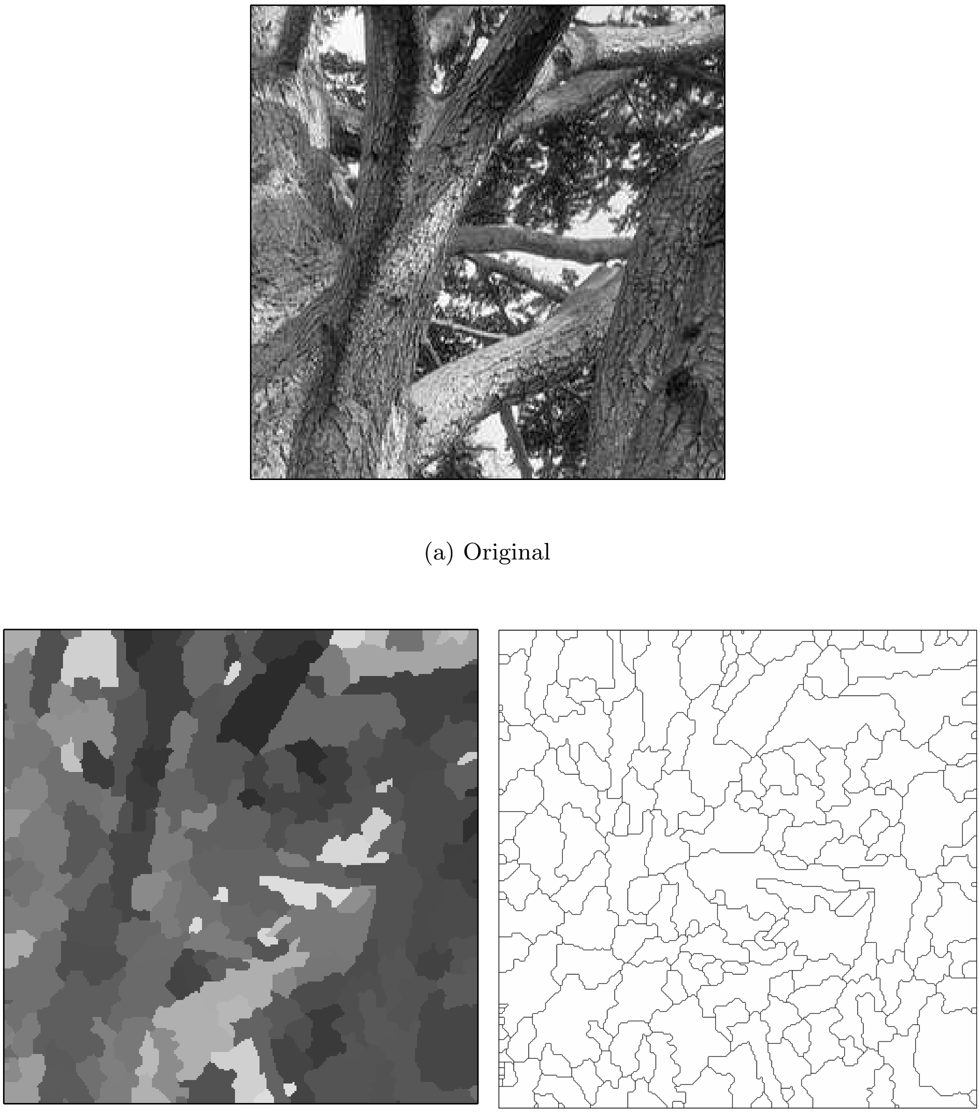
(c) $\alpha = 0.5$ Outline.



(d) $\alpha = 0.75$ Mean replaced.

(e) $\alpha = 0.75$ Outline.

Figure E.4: The original image of 'toys' (a) and the mean replaced and outline images for $\alpha = 0.5$ and 0.75 (b) to (e).



(a) Original

(b) $\alpha = 0.6$ Mean replaced.

(c) $\alpha = 0.6$ Outline.

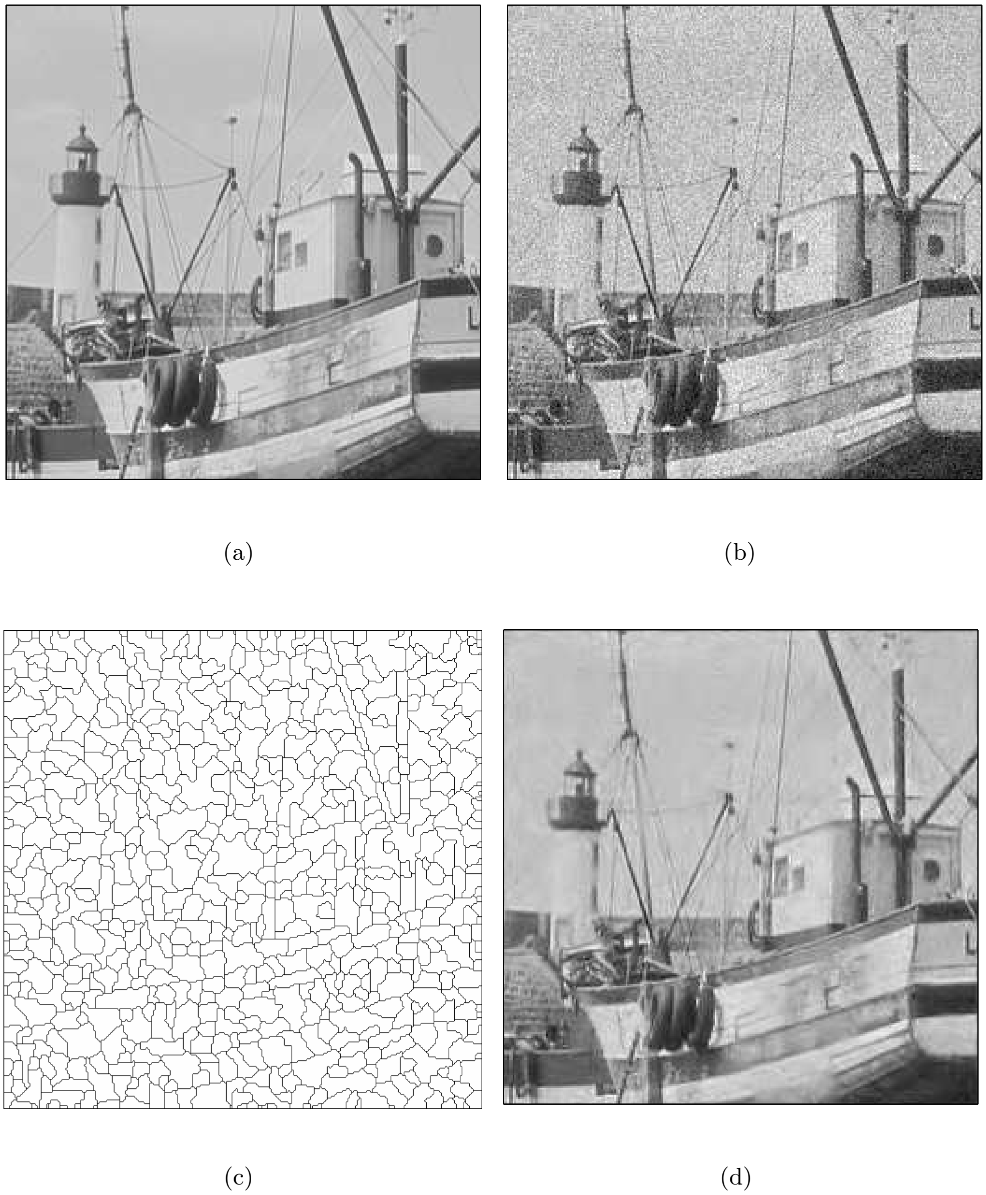
Figure E.5: The original image of 'tree' (a) and the mean replaced and outline images for $\alpha = 0.5$ (b) and (c).

Appendix F

Additional Results of Wavelet Based Noise Removal

This appendix presents further results of the wavelet based noise reduction scheme presented in Section 5.3 using both the adaptive λ-max gradient smoothing and adaptive λ-max open-closing / grey scale-merging segmentation strategies. All images are contaminated by Gaussian white noise of standard deviation 15 prior to segmentation and subsequent noise reduction.

F.1 Adaptive λ-max Gradient Smoothing Segmentation



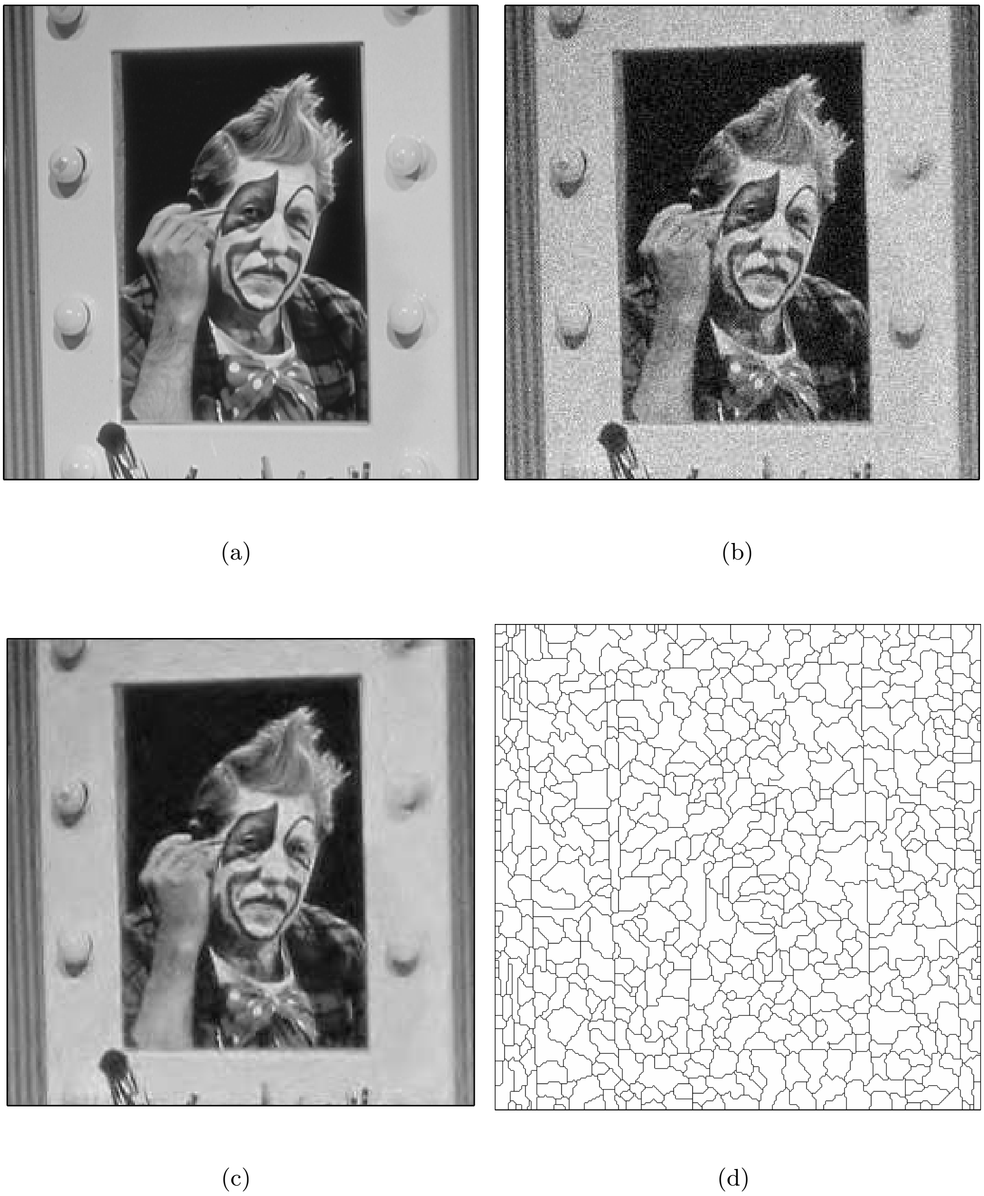
(a)

(b)

(c)

(d)

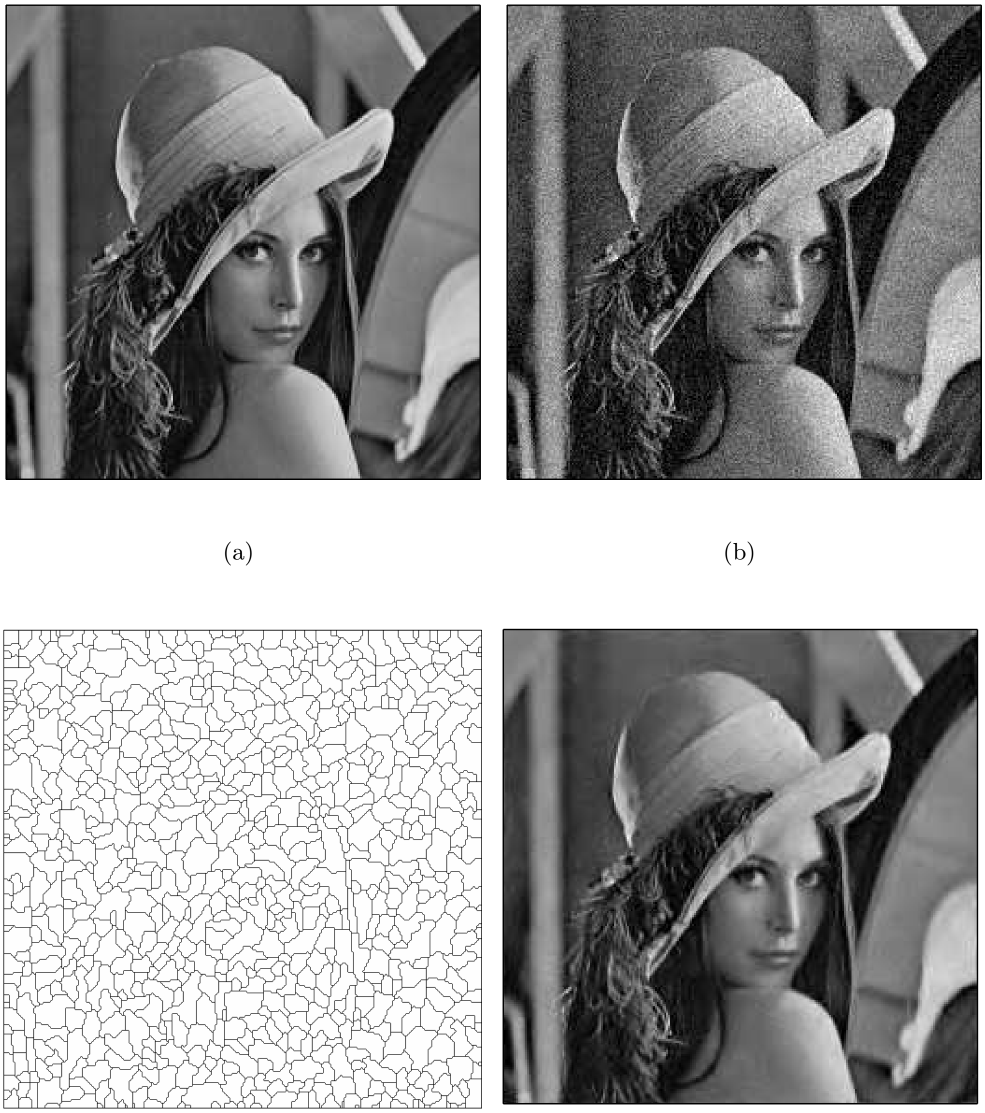
Figure F.1: The original image of ‘boat’ (a) the contaminated image (b) and the outline (c) and enhanced images (d) corresponding to the ‘best’ noise reduction.



(a) (b)

(c) (d)

Figure F.2: The original image of ‘clown’ (a) the contaminated image (b) and the outline (c) and enhanced images (d) corresponding to the ‘best’ noise reduction.



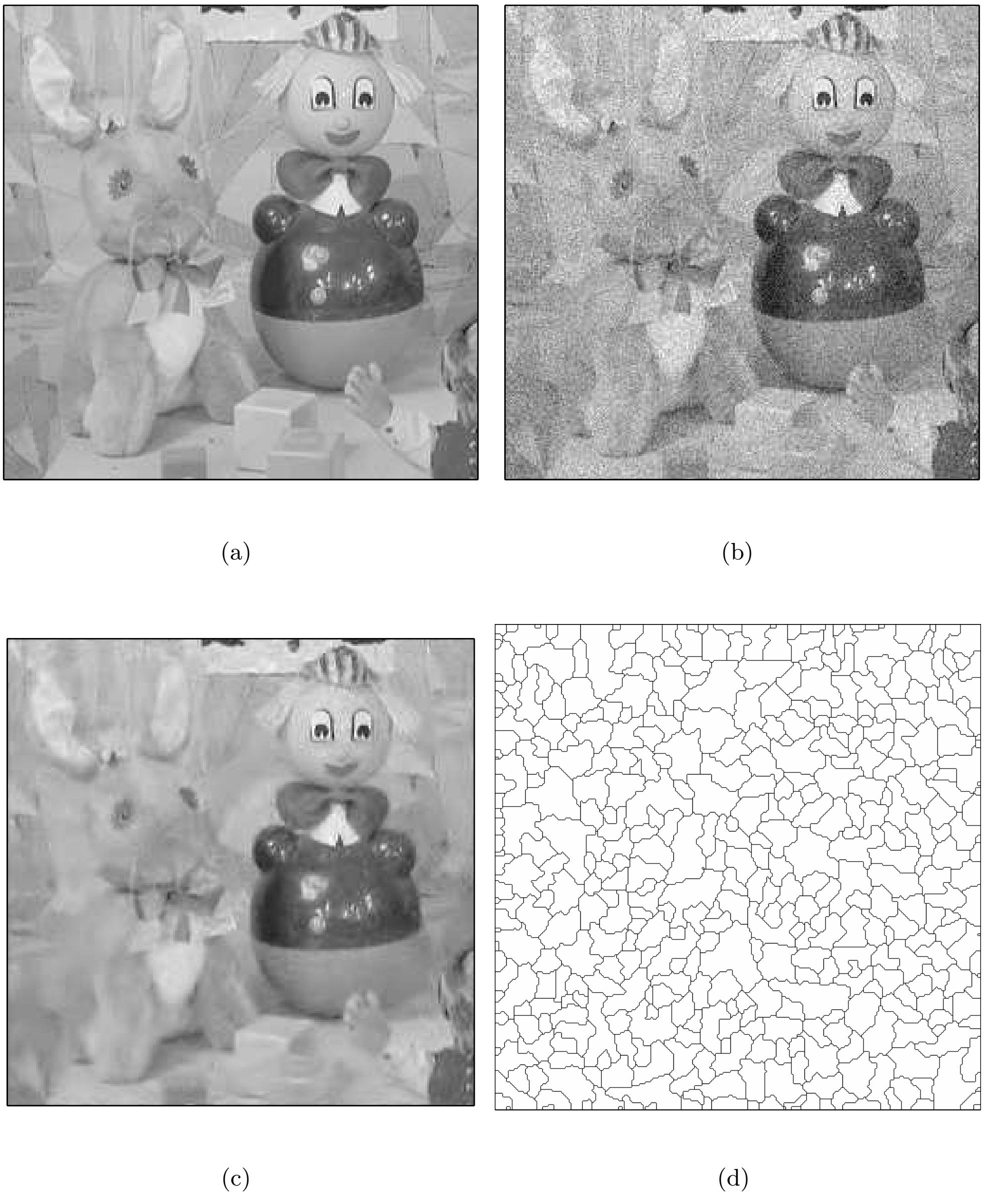
(a)

(b)

(c)

(d)

Figure F.3: The original image of 'lenna' (a) and the outline (b) and enhanced images (c) corresponding to the 'best' noise reduction.



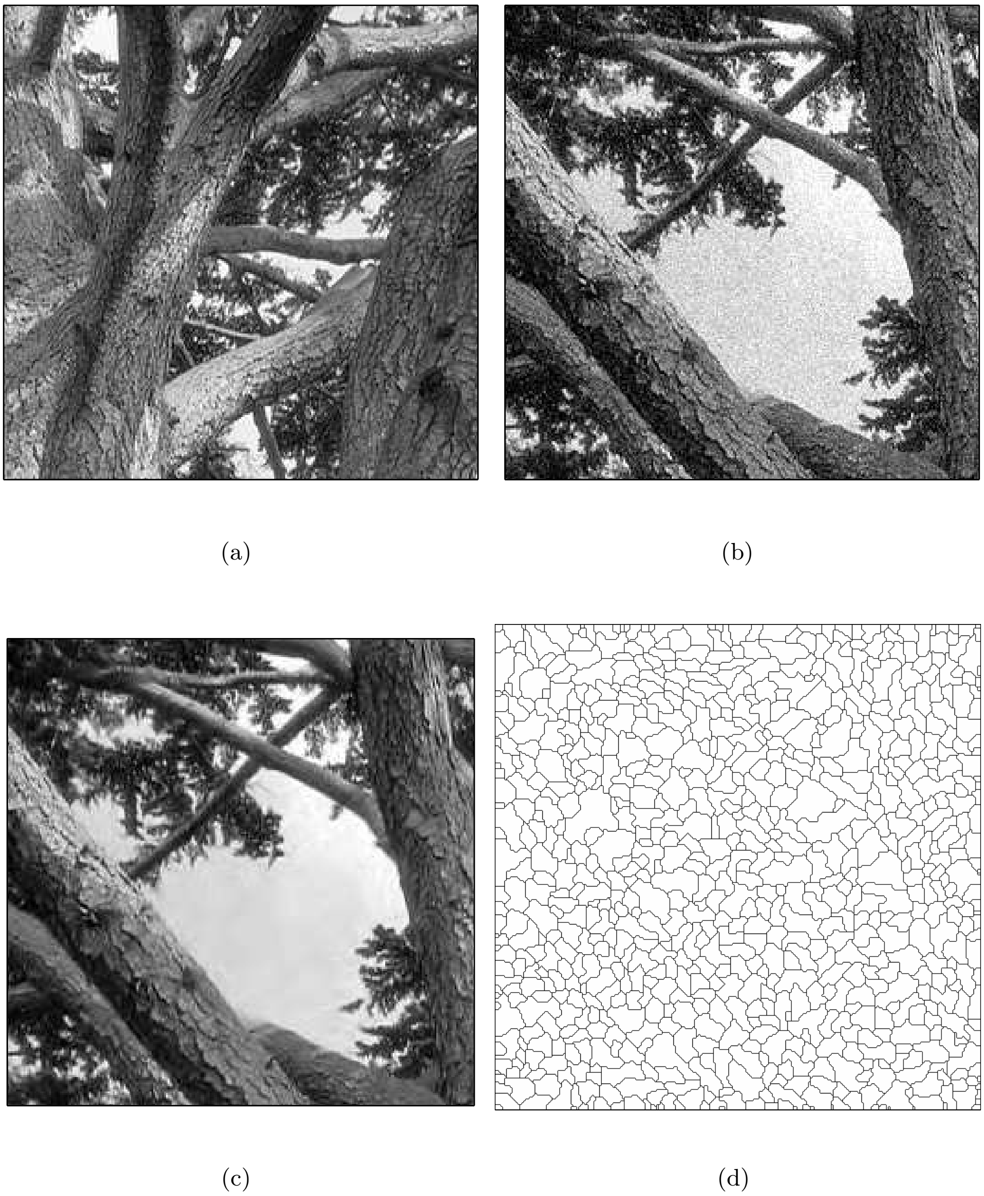
(a)

(b)

(c)

(d)

Figure F.4: The original image of ‘toys’ (a) the contaminated image (b) and the outline (c) and enhanced images (d) corresponding to the ‘best’ noise reduction.



(a) (b)

(c) (d)

Figure F.5: The original image of ‘tree’ (a) the contaminated image (b) and the outline (c) and enhanced images (d) corresponding to the ‘best’ noise reduction.

References

[1] Sonka, Hlavac, and Boyle, *Image Processing, Analysis and Machine Vision.* PWS Publishing, 2^{nd} ed., 1999.

[2] R. Haralick and L. Shapiro, “Image segmentation techniques,” *Computer Vision, Graphics and Image Processing*, vol. 29, pp. 100–132, 1985.

[3] P. Scheunders and J. Sijbers, “Multiscale watershed segmentation of multivalued images,” *Proc. ICPR02,* 16^{th} *International Conference on Pattern Recognition, Quebec*, pp. 679–688, 2002.

[4] D. Marr and E. Hildreth, “Theory of edge detection,” *Proceedings of the Royal Society*, vol. B 207, pp. 187–217, 1980.

[5] R. Gonzalez and P. Wintz, *Digital Image Processing.* Addison-Wesley, 1977.

[6] M. H. F. Wilkinson and J. B. T. M. Roerdink, “Fast morphological attribute operations using tarjan’s union-find algorithm.,” *in Mathematical Morphology and its Applications to Image and Signal Processing*, pp. 311–320, 2000. J. Goutsias, L. Vincent and D.S. Bloomberg, Eds.

[7] S. T. Acton, “Fast algorithms for area morphology,” *Digital Signal Processing*, vol. 11, no. 3, pp. 187–203, 2001.

[8] A. Bleau, J. D. Guise, and R. LeBlanc, “A new set of fast algorithm for mathematical morphology ii. identification of topographic features on greyscale images,” *Computer Vision, Graphics and Image Processing: Image Understanding*, vol. 56, pp. 210–229, September 1992.

[9] A. Bleau, J. D. Guise, and R. LeBlanc, “A new set of fast algorithm for mathematical morphology i. idempotent geodesic transforms,” *Computer Vision,*

Graphics and Image Processing: Image Understanding, vol. 56, pp. 178–209, September 1992.

[10] R. Haralick, S. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Trans. Pattern Analysis and Machine Vision*, vol. PAMI-9, pp. 532–550, July 1987.

[11] P. Maragos and R. Schafer, "Morphological filters - part i: Their set theoretic analysis and relations to linear shift-invariant filters," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. ASSP-35, pp. 1153–1169, August 1987.

[12] P. Maragos and R. Schafer, "Morphological filters - part ii: Their relations to median, order statistic and stack filters," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. ASSP-35, pp. 1170–1184, August 1987.

[13] J. Serra, *Image analysis and mathematical morphology.* Academic press, 1982.

[14] P. Maragos, "Tutorial on advances in morphological image processing and analysis," *Optical Engineering*, vol. 26, pp. 623–632, July 1987.

[15] D. Gatica-Perez, C. Gu, M.-T. Sun, and S. Ruiz-Correa, "Extensive partition operators, gray-level connected operators and region merging / classification segmentation algorithms: Theoretical links," *IEEE Transaction on Image Processing*, vol. 10, pp. 1332–1345, September 2001.

[16] G.-I. Lee, I.-K. Kim, D.-W. Jung, J.-H. Song, and W.-G. K. nad Rae-Hong Park, "Watershed based edge detection in noisy images," *Proceedings of IASTED International Conference on Signal and Image Processing*, pp. 469–473, August 2002. Hawaii.

[17] F. Meyer and S. Beucher, "Morphological segmentation," *Journal of Visual Communication and Image Representation*, vol. 1, pp. 21–46, Sept 1990.

[18] J. B. T. M. Roerdink and A. Meijster, "The watershed transform: Definitions, algorithms and parallelization strategies," *Fundamenta Informaticae*, vol. 41, no. 1-2, pp. 187–228, 2000.

[19] F. Meyer and C. Vachier, "Image segmentation based on viscous flooding simulation," *Proc. of the VIth International Symposium of Mathematical Morphology*, pp. 69–77, April 2002.

[20] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, 1991.

[21] P. Felkel, M. Bruckschwaiger, and R. Wegenkittl, "Implementation and complexity of the watershed-from-markers algorithm computed as a minimal cost forest," *Computer Graphics Forum*, vol. 20, pp. 26–35, September 2001.

[22] K. Haris, S. Efstratiadis, N. Maglaveras, and A. Katsaggelos, "Hybrid image segmentation using watersheds and fast region merging," *IEEE Transactions on Image Processing*, vol. 7, no. 12, pp. 1648–1699, 1998.

[23] J. Gauch, "Image segmentation and analysis via multiscale gradient watershed hierarchies," *IEEE Transactions on Image Processing*, vol. 8, pp. 69–79, January 1999.

[24] D. Wang, "A multiscale gradient algorithm for image segmentation using watersheds," *Pattern Recognition*, vol. 30, no. 12, pp. 2043–2052, 1997.

[25] L. Garrido, P. Salembier, and D. Garcia, "Extensive operators in partition lattices for image sequence analysis," *Signal Processing*, vol. 66, pp. 157–180, April 1998.

[26] H. J. Heijmans, *Morphological Image Operators*. Academic Press Inc., 1994.

[27] J. A. Bangham, T. G. Campbell, and R. V. Aldridge, "Multiscale median and morphological filters for 2d pattern recognition," *Signal Processing*, vol. 38, pp. 387–415, August 1994.

[28] F. Cheng and A. N. Venetsanopoulos, "An adaptive morphological filter for image processing," *IEEE Transactions on Image Processing*, vol. 1, pp. 533–539, October 1992.

[29] L. Vincent, "Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms," *IEEE Transaction on Image Processing*, vol. 2, pp. 117–201, April 1993.

[30] P. Salembier and J. Serra, "Flat zones filtering, connected operators, and filters by reconstruction," *IEEE Transactions on Image Processing*, vol. 4, pp. 1153–1160, August 1995.

[31] R. L. Stevenson and G. R. Arce, “Morphological filters: Statistics and further syntactic properties,” *IEEE Transactions on Circuits and Systems*, vol. CAS-34, pp. 1292–1305, November 1987.

[32] E. Breen and R. Jones, “Attribute openings, thinnings and granulometries,” *Computer Vision and Image Understanding*, vol. 64, pp. 377–389, 1996.

[33] N. Young and A. N. Evans, “Digital video pre-processing with multi-dimensional attribute morphology,” *Proceedings of IEE Visual Information Engineering 2003*, July 2003.

[34] J. Serra and P. Salembier, “Connected operators and pyramids,” *In SPIE Image Algebra and Mathematical Morphology*, vol. 2030, pp. 65–76, July 1993. San Diego (CA) USA.

[35] A. Meijster and M. Wilkinson, “A comparison of algorithms for connected set openings and closings,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 484–494, April 2002.

[36] A. Oliveras and P. Salembier, “Generalized connected operators,” *Visual Communication and Image Processing, VCIP’96*, pp. 761–773, March 1996.

[37] S. T. Acton and D. P. Mukherjee, “Area operators for edge detection,” *Pattern Recognition Letters*, vol. 21, pp. 771–777, 2000.

[38] S. T. Acton and D. P. Mukherjee, “Scale space classification using area morphology,” *IEEE Transactions on Image Processing*, vol. 9, pp. 623–635, April 2000.

[39] P. Salembier, P. Brigger, and J. C. nad Montse Paradás, “Morphological operators for image and video compression,” *IEEE Transactions on Image Processing*, vol. 5, pp. 881–898, June 1996.

[40] P. Salembier, A. Oliveras, and L. Garrido, “Antiextensive connected operators for image and sequence processing,” *IEEE Transactions on Image Processing*, vol. 7, pp. 555–570, April 1998.

[41] P. Salembier and A. Oliveras, “Practical extensions of connected operators,” *Mathematical Morphology and its applications to Image and Signal Procesing, ISMM’96*, pp. 97–110, May 1996. Atlanta, GA, USA.

[42] L. Garrido, P. Salembier, and A. Oliveras, "Anti-extensive connected operators with application to image sequences," *VII Simposium Nacional de Reconocimiento de Fromay y Análisis de Imágenes*, vol. 1, pp. 151–156, April 1997. Barcelona Spain.

[43] L. Garrido, A. Oliveras, and P. Salembier, "Motion analysis of image sequences using connected operators," *SPIE Visual Communication and Image Processing*, vol. 3024, pp. 546–557, Feb 1997. San Jose California USA.

[44] R. Roman-Roldan, J. F. Gomez-Lopera, C. Atae-Allah, J. Martinez-Aroza, and P. L. Luque-Escamilla, "A measure of quality for evaluating methods of segmentation and edge detection," *Pattern Recognition*, vol. 34, pp. 969–980, May 2001.

[45] L. Yang, F. Albergtsen, T. Lonnestad, and P. Grottum, "A supervised approach to the evaluation of image segmentation methods," *Proceedings of the 6th International Conference on Computer Analysis of Images and Patterns*, pp. 759–765, September 1995. Prague, Czech Republic.

[46] Y. J. Zhang, "A survey on evaluation methods for image segmentation," *Pattern Recognition*, vol. 29, pp. 1335–1346, August 1996.

[47] M. Everingham, H. Muller, and B. Thomas, "Evaluating image segmentation algorithms using monotonic hulls in fitness / cost space," *Proceedings of the 12th British Machine Vision Conference (BMVC2001)*, 2001.

[48] C. E. Erdem and B. Sankur, "Performance evaluation metrics for object-based video segmentation," *Proceedings of the 10th European Signal Processing Conference (EUSIPCO 2000)*, pp. 917–920, September 2000. Tampere, Finland.

[49] C. Shaffrey, I. Jermyn, and N. Kingsbury, "Psychovisual evaluation of image segmentation algorithms," *Proceeding of Advanced Concepts for Intelligent Visual Systems.*, September 2002. Ghent, Belgium.

[50] J. Liu and Y.-H. Yang, "Multiresolution color image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, July 1994.

[51] M. Borsotti, P. Campadelli, and R. Schettini, "Quantitative evaluation of color image segmentation results," *Pattern Recognition Letters*, vol. 19, pp. 741–747, June 1998.

[52] J. Rissanen, “Modelling by shortest data description length,” *Automatica*, vol. 14, pp. 465–471, 1978.

[53] J. Rissanen, “A universal prior for the integers and estimation by the minimum description length,” *Annals of Statistics*, vol. 11, pp. 416–431, 1983.

[54] A. Kokaram, *Motion Picture Restoration.* Springer Verlag, 1998.

[55] A. DeStefano, *Wavelet-Based Reduction of Spatial Video Noise.* Ph.d thesis, University of Southampton, 2000.

[56] S. G. Chang, B. Yu, and M. Vetterli, “Spatially adaptive wavelet thresholding with context modeling for image denoising,” *IEEE Transactions on Image Processing*, vol. 9, pp. 1522–1531, September 2000.

[57] A. Chambolle, R. A. DeVore, N. yong Lee, , and B. J. Lucier, “Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage,” *IEEE Transactions on Image Processing*, vol. 7, pp. 319–335, March 1998.

[58] D. L. Donoho, “De-noising by soft-thresholding,” *IEEE Transactions on Information theory*, vol. 41, no. 3, pp. 613–627, 1995.

[59] R. R. Coifmann and D. L. Donoho, “Translation invariant de-noising,” *in Wavelets and Statistics*, 1995. Springer-Verlag Ed, A. Antoniadis.

[60] P. Hill, N. Canagarajah, and D. Bull, “Image segmentation using a texture gradient based watershed algorithm,” *IEEE Transaction on Image Processing*, vol. 12, pp. 1618–1633, December 2003.

[61] J. Breuer, *The Theory of Sets.* Prentice Hall Inc., 1958.

[62] J. B. Fraliegh, *A First Course in Abstract Algebra.* Addison-Wesley Publishing Company, Inc., 6$^{\text{th}}$ ed., 1999.