Colour Image Quantisation and Coding

For Optimal Perception

Roderick William McColl, BSc(Hons), MSEE

A thesis submitted to

The University of Warwick

for the degree of

Doctor of Philosophy

Department of Computer Science

June 1991.

Abstract

Once a digital image is processed in some way and the reconstruction is compared to the original, the final arbiter of reconstruction quality is the human to whom the images are presented. The research presented here is concerned with the development of schemes for the quantisation of colour images and for the encoding of colour images for transmission, with the goal of minimising the perceived image distortion rather than minimising a traditional error signal statistic.

In order to quantise colour images with minimum perceived distortion, a colour space is sought in which Euclidean distances correspond linearly to perceived colour difference. The response of the visual system to colour and colour difference is investigated. A new quantisation scheme is developed and implemented to achieve a colour image compression ratio of approximately 6:1. Three variations on the basic quantiser algorithm are considered and results of applying each variation to three test images are presented.

Two-component encoding of colour images for low bit-rate transmission is investigated. A new method of encoding the contents of the image regions following contour extraction is developed. Rather than using parametric surface descriptions, a quad-tree is constructed and a simple measure of perceived image contrast threshold is used to determine the transmitted data. Arithmetic entropy coding is used to discard statistical redundancy in the signal. A colour wash process recreates the colour in each region. Implementation details are presented and several examples are given to illustrate differing contrast thresholds with compression rates of up to 50:1.

An analysis of the textures in certain regions of the test images leads to the development of an algorithm to synthesise the appearance of the textures following extraction of a small block which may be repeated across the region, leading to dramatic compression rates in some instances.

List of Contents

List of Figures

List of Tables

Declaration

I hereby declare that, except where specifically acknowledged, all work presented in this thesis was carried out by me, Roderick William McColl, and that it has not been submitted elsewhere for the purpose of obtaining an academic degree.

Acknowledgements

There are a number of individuals to whom I would like to express my deepest gratitude. I thank my supervisor, Dr. Graham Martin, for his considerable help both during my years at Warwick and also during the period of writing and submitting the thesis from Texas. Dr. Roland Wilson gave me much technical advice on image coding; my fellow students (now Drs.) Andrew Calway and Martin Todd also provided many valuable suggestions. My colleagues at GEC Hirst Research Centre, Dr. Vaughan Stanger and Ms. Alexandra Symons, involved me in many helpful discussions. Finally I thank my parents who have always encouraged me to complete whatever academic endeavour I start.

This research was carried out under the collaborative Alvey Project MMI085, 'Image Analysis and Coding.' I am grateful to the Alvey Directorate and the UK Science and Engineering Research Council for making these funds available.

Chapter 1. Introduction

This thesis concentrates on aspects of colour image quantisation, redundancy and coding, and how the pertinent factors may be manipulated within the context of minimising the perception of noise or errors by the average viewer of processed images; by using perceptual criteria, any processed image should contain an "optimal" distortion, i.e. one that is least easily perceived, hence the thesis title. The investigation of these factors includes visits to the fields of human physiology, colour science, optical physics, image processing and information theory. The primary motivation in the investigation is the considerably increased raw storage requirement of typical colour images compared with their grey-scale counterparts, and the likely impact of successful compression of the image beyond that which is currently considered acceptable. A specific application for such compression would be video conferencing at the increasingly available ISDN (64kbit/s x N) bandwidths.

1.1. Theories of Vision

The central concept of colour vision is that of *trichromacy*, which holds that colour vision is a function of at least three variables. It is widely held that this theory was proposed in 1807 by Thomas Young [1] and elaborated by von Helmholtz [2] a half century later (see MacAdam [3] for evidence to the contrary.) Young assumed that the eye contains three independent response mechanisms to light: one predominantly sensitive to the shorter wavelengths of light, the second favouring the middle part of the visible spectrum and the third most sensitive to the longer wavelengths. Each of these responses would have some value over the whole visible spectrum, and the visual sensation to a given light source would come from the integration of the three responses with the components of the light. In 1860 Maxwell determined the shape of three responses from experimental observations [4] (a reprint of Maxwell's paper can be found in MacAdam [5]). Maxwell's experiment is widely considered to be the first fundamental study of colour vision. His experiment involved the arbitrary choice of three spectral wavelengths or primaries, from which light of any wavelength or wavelengths could be generated, and subsequently matched, or distinguished, by the receptors of the eye. These primaries can be labelled "red", "green" and "blue" according to the subjective appearance of monochromatic light of the selected primary wavelengths. Maxwell selected the primary wavelengths to be 630 nm for the red primary, 528 nm for the green primary and 457 nm for the blue primary. These colour receptors in the retina are named "cones", because of their physical appearance.

Competing theories to Young's soon appeared [4]. The motivation for alternative theories for colour vision lay in the way the previous models performed in the case of abnormal colour vision, such as colour blindness. The theory did not adequately account for the sensations experienced by dichromats, the most common of which is the inability to distinguish between red and green. Konig [6] suggested that one of the colour receptors in the eye may actually perceive brightness only, i.e possessed an achromatic response, with the other two mechanisms providing the chromatic response. Hering [7] proposed a theory based on these ideas in 1878, known as the *opponent-colours* theory. The theory states that the sensation of colour is a result of three opposing pairs of processes: a light-dark, a red-green and a yellow-blue. From a physiological standpoint, this theory allowed for dichromatic vision simply by the notion of the loss of one of these processes. Unfortunately, the advantage in Young's theory was that it was simple and deemed physiologically plausible, whereas Hering's theory was considered unacceptable physiologically.

The logical progression from these early theories were more complex models of colour vision which included parts of both trichromacy and opponent-colours [4]. These are known as *stage* theories because they incorporate one or more distinct transformations of the processes in the visual system, from the retina to the optic nerve. The stages began with a retinal process similar to that suggested by Young, ending with a signal similar to an opponent-colours process. The Young model became by far the more popular, and little research was undertaken to provide a quantitative account of opponent-colours processes. Although Young's cone theory has always been generally accepted, however, it was not until the latter part of this century that any physical evidence for such responses appeared in the literature [8].

1.2. Colour Standards

As the science of colour vision developed, many researchers contributed their own models, each differing slightly in an attempt to account for those situations in which other models failed. In order to provide some sort of international standard, in 1931, the Commission Internationale de l'Eclairage (CIE) [9] defined the colour response curves for the eye of the *Standard Observer*, a normal male trichromat (in fact an average of previous experiments by Wright [10] and Guild [11]), in terms of spectrum primaries of 600 nm, 546 nm and 436 nm, considered to be a better choice of primaries from a physiological standpoint. These curves have been used to specify paints, television monitors, etc. from the view of how colours appear to a normal human. These curves are shown in figure 1.1 (from Judd [4]). It is apparent that the range of light that the human visual system can perceive is about 380 to 700 nm.

As well as developing standardised physiological models, the CIE is also the governing body for colour space definitions [9]. A colour space is defined as that volume occupied by all possible colour vectors with axes drawn from the three selected primaries. One of the disadvantages of the normal colour matching functions using the red, green and blue primary wavelengths, is that in certain situations, negative amounts of one or more primaries are required to generate the desired colour [4]. Obviously there can be no such thing as a negative amount of radiant flux; however, the concept is well understood to mean that some amount of one of the primaries is subtracted from the light incident on the retina in order to make the colour match (this is one of the central concepts in the laws of Grassman for compound colours [12]). However, since an additive colour system is a requirement for the likes of colour display systems such as televisions, the CIE established a colour space in which all colours that can be seen are composed of positive amounts of the three primaries. This colour space is referred to as the (CIE) XYZ colour space [9], and it contains what is known as the cone of realisable colours.

A more common representation of vectors in the CIE colour space is the specification of the chromaticity of a colour, which, together with a third value, the luminance, (Y), provide a direction and magnitude from the origin. Working with a 2-dimensional chromaticity space makes the graphical display of graphical colour information more straightforward, and also provides a dimensionless, normalised, reference for the spectrum, which resolves the extreme changes in energy found at different light wavelengths.

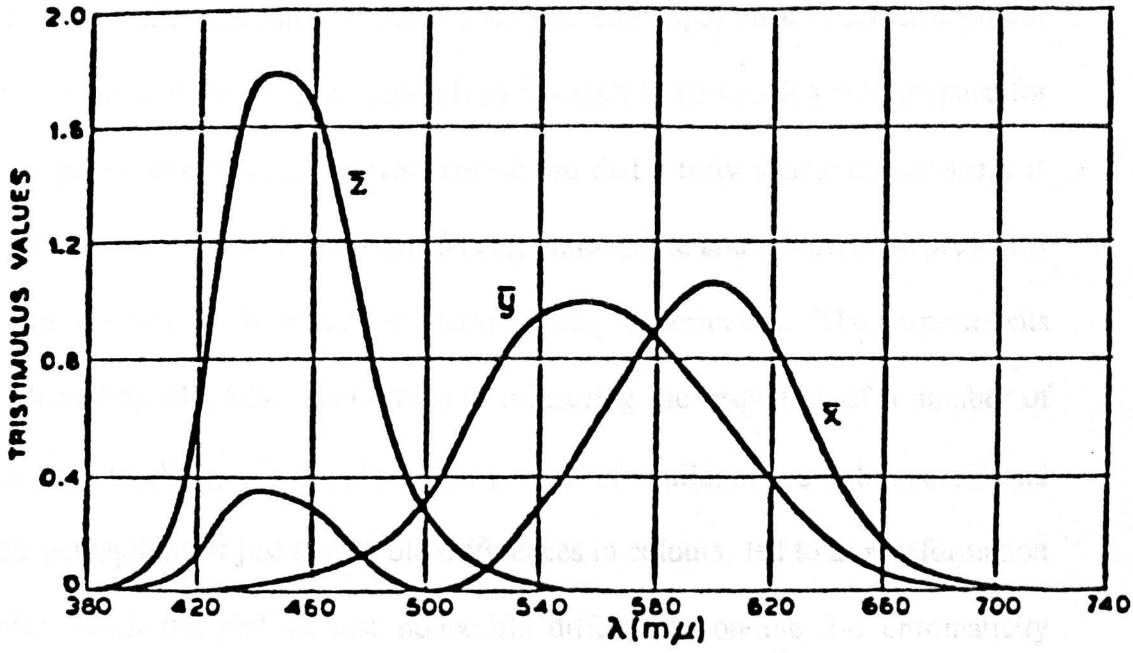


Figure 1.1. CIE colour matching functions for the Standard Observer.

(Source: Judd [4]).

1.3. Colour Spaces

It has long been recognised that there is considerable redundancy in the use of a colour space developed from the red, green and blue primaries. As well as the fact that the eye's response to the red sensation is very similar to the green sensation, the human observer is much more sensitive to changes in the middle and longer wavelengths than to the shorter ones (see Wald [8] for example). This means that as the spectral content of the generated light is perturbed, the colour difference perceived by the observer is not isotropic. The colour space is not metric from the viewpoint of sensed changes. The importance of a metric colour space to the printing industry has driven research to find such a colour space for many years. MacAdam, however, has shown that a truly metric colour space is impossible to attain in 3 dimensions [13], although he and others have presented approximations to it based on curve fitting experiments. The experiments conducted by MacAdam and others in measuring the responses of a number of observers to changes in the chromatic content of equiluminous light sources and their perceptions of just noticeable differences in colours, led to a transformation under which the plot of just noticeable differences on the 2-d chromaticity diagram produced circles. Plotting within the CIE (x,y) chromaticity space or the (r,g) chromaticity space show ellipses, with longer axes in the direction of the green and red primaries. MacAdam was unsuccessful in getting the CIE to adopt his geodesic chromaticity model [14], so called because the curve between any two colours on the chromaticity diagram is a straight line, or *geodesic*, and involves the least number of colour differences. However the CIE in 1961 adopted his recommendations for a more metric space than the XYZ colour space, labelled the UVW colour space [15]. Transformations from RGB to XYZ and to UVW can all be achieved through matrix multiplication.

Other colour spaces which have been proposed include the Lab space [16], the HSL (Hue, Saturation, Lightness) space [17] and the Munsell system [18], which was conceived by the artist Edward Munsell in 1905 as an attempt to bring a rigourous language to the description of colours. The Munsell Colour Company's tables of colour tiles are used extensively in the colour industry today.

During the 1950s and 1960s, Hurvich and Jameson conducted a number of investigations within the framework of an opponent-colours theory [19]. Their argument for this model was that an enormous amount of research had been expended in finding the "best" set of primary wavelengths under a Young model, with no general consensus emerging as to the "right" choice, because of the failure of the model to account for all the facts of colour vision. Their work yielded fundamental quantitative models whereas, before, the data were decidedly qualitative. In fact, their model was a two-stage model, incorporating a trichromatic "front end" to the visual system with response curves similar to that of the CIE standard observer, and an appropriate transformation to yield the light-dark, red-green and yellow-blue sensations which are transmitted along the optic nerve. They stated that physiologically, this model made excellent sense because it explained, in a simple fashion, phenomena such as chromatic adaptation and colour constancy, for which the three colours theories failed to provide. Ohta *et al.* [20] also propose, from empirical observations, an opponent-colours model in which three best-fit colour response functions are used to segment colour images.

The opponent-colours theory is of interest because the implication is that the visual system performs its own colour compression [21], as the red, green and blue responses from the cones in the eye are transformed into a space with three pathways which contain high-bandwidth information (the light-dark sensation), medium-bandwidth information (the red-green sensation) and low-bandwidth information (the yellow-blue sensation). In recent years, many advances in image processing have been made by considering human physiology and attempting to exploit its weaknesses, rather than generating ad hoc models and algorithms. It seems to make sense, then, to base investigations of colour image properties on considerations of the characteristics of the visual system, since that is the intended "receiver".

1.4. Image Processing

There are many examples in the literature of quantisation schemes for colour image data. Since the work of Max on quantisation with a minimum mean-square error criterion [22], and the subsequent work of Linde, Buzo and Gray on vector quantisation [23], there have also been attempts to perform space decomposition using the Mandelbrot set [24,25] and a variety of clustering and binning algorithms designed to find a good representative set of image colours. There is, however, little work in the literature on the subject of the choice of a suitable colour space in which to quantise colour data, although the review by

Limb et al. [26] does discuss this issue.

Image processing has evolved rapidly as a science throughout its reasonably brief history. The main stumbling block to applications has been the sheer data throughput required when developing and testing algorithms, a problem which has only recently eased with the development of inexpensive, fast computer technology. Two distinct generations of image processing algorithms have been reported in the literature [27]. The so-called first generation includes both spatial and transform techniques, for instance DPCM and DCT algorithms, both of which have been extensively represented in the literature, but the majority of algorithms have at best assumed that colour image processing is no more than an extension of gray-scale processing, with the two chromatic channels simply being assumed to occupy narrower bandwidths than the achromatic signal, but possessing similar properties.

The second-generation image processing algorithms, which in fact date back to Schreiber's "Synthetic High" processing model [28], attempt to account for or model, at varying levels of resolution and complexity, the framework and behaviour of the visual system. They are known as *two-component* image models. There are a number of possible categories that these models fall into, such as pyramid coding and texture coding [27]. Pyramid coding schemes attempt to take into account the fact that the visual system incorporates parallel filtering processes at differing resolutions. Texture coding schemes attempt to find simple base patterns which can be repeated throughout a region of the image to synthesise the original. The model which has been investigated in this research is known as the *region-growing* model. The model assumes that the average scene or image frame, is composed of large areas or regions in which the data is stationary and in which the variation in pixel value or values is small. These regions are bounded by discontinuities, known as edges, where there are abrupt changes in the image content. The processing of the two feature types involves encoding the edge pixels and finding succinct descriptions of the slowly changing information in the regions. The human visual system is known to be far more sensitive to discontinuities in image content than to slowly changing properties, and the expected compression from this type of coding scheme is potentially enormous.

1.5. Image Distortion

Another area of interest within this investigation has been the determination of a suitable error measure for the results of image processing algorithms. The traditional mean-square error estimate is widely agreed to be a poor representative of the success of any image compression scheme [26,29,30], simply because it is physiologically unacceptable, requiring the visual system to calculate square roots, as well as providing only global statistics, which do not accurately follow the few large distortions which may be found in one image and not in another with the same measure. Referring to the last paragraph, it is known that the visual system is far more sensitive to sharp changes in an image than to low spatial-frequency ones. An ability to estimate the impact of a small number of large errors within a processed image must be taken into account. This thesis includes analyses of the applicability of error measures to the success of colour image quantisation and coding schemes with a view to providing substantial agreement between subjective appearance and numerical error value.

1.6. Other Chapter Contents

The next chapter contains an investigation of colour quantisation for minimum perceived distortion, with a further description of the colour space and chromaticity space, colour space homogeneity, linearity and metricity. Quantisation schemes which are found in the literature are discussed. A novel quantisation is developed, along with complete details of an implementation. Results are obtained from three test images, together with a discussion of error statistics. A comparison of the success of the algorithms developed in this chapter with selected other literature is also given.

In chapter three the role of colour within the region-growing image coding model is discussed. A survey of image coding techniques found in the literature is given, with emphasis on two-component coding concepts. This is followed by a brief presentation of the methods used to generate the edges and regions which make up each image frame. The rest of the chapter is devoted to an investigation of models for coding colour regions and their applicability within the framework of the human visual system. The details of a new region-coding model are presented, along with an investigation of arithmetic entropy-coding schemes for further compression of the region descriptions. Implementation details and results are presented in depth, and comparisons are made with other literature.

Chapter four is concerned with the analysis and synthesis of the textures which can be identified in colour images, and the potential for compression based on popular texture models in the literature. An algorithm is developed to improve the compression achieved in chapter three on certain colour image regions containing a feature which is regularly or semi-regularly repeated; the synthesis of such features and the rules for repetition are derived and a comparison is made between cost and perceived quality of reconstructed regions using the algorithms developed in this chapter and in the previous one.

Chapter five includes summaries of chapters two, three and four, discussions of those areas in which further improvements could be made, a conclusion of the results obtained overall and proposals for further research in the field of colour image processing.

Chapter 2. Quantisation of Colour Image Data

Consider the storage requirements for a typical digitised colour image. The matrix size is usually 256x256 pixels at low resolution, and perhaps 512x512 pixels for a high-quality image (most television standards offer resolutions in the 500-600 line range). If the data are stored as RGB triplets, the typical quantisation is to 8-bits per primary, 24-bits per pixel, or 24 bpp. Therefore, for a high-resolution digitised colour image, the requirement is 786432 bytes, or three-quarters of a MegaByte. Even in the days of inexpensive magnetic media, a sequence of such images requires a considerable storage commitment, and access to any one frame or pixel for real-time applications requires high-speed disk access hardware. Given that the human observer is incapable of distinguishing the 16.7 million colours available at 24-bpp, it is often desirable to provide quantisation to considerably reduce the storage cost per frame, depending on the application envisaged for the data.

In this chapter, we will investigate an algorithm, designed to find the quantisation of any colour image such that any introduced distortion in the reconstructed image is always the least easily perceived. Three variations on a basic quantisation algorithm are presented, together with a number of reconstructed images, and an attempt is made to quantitatively measure the distortion, if any, that is introduced by the quantisation process. The main interest is in reducing as far as possible the cost of storing the chromatic content of the colour image. The results are favourable when compared with those found elsewhere in the literature.

2.1. Introduction

Since the early 1970s, there has been interest in processing colour images as well as monochromatic ("black-and-white") images. Quantisation of the image "signal" is generally implicit in the processing algorithm. In the 1980s, the burgeoning field of computer graphics produced much research on quantities such as the most suitable number of colours to use on display and the science of ergonomics has produced much research on the interaction of the observer with colour information on his display screen. Since typical displays allow 256 colours to be displayed concurrently, algorithms to generate 8-bit colour maps from 24-bit images are prevalent in the literature. The work of Stockham [31] (for monochromatic signals) and Faugeras [29] highlighted the importance of working with a model of human perception. Faugeras' vision model works as an opponent-colours process and, in the case of achromatic signals, reduced to that of Stockham's. He used empirical studies to compute matrix coefficients to transform experimental cone absorption spectra into his own colour space, and demonstrated an agreement between subjective measurements of introduced colour image distortions with MSE measurements for the distorted images. He did not attempt any quantisation experiments, however. Stenger [32] considered the perception of chrominance differences in his design for a quantiser of television signals. He computed a number of luminance-dependent grids to quantise the signal. Kurz [17] used a similar scheme and incorporated the MacAdam geodesic as a quantisation grid for pixel chromaticities, rather than the chrominance considered by Stenger. Multiple grids were used depending on the quantised luminance value. Many schemes which have become popular, however, such as those of Heckbert [33], and Lena and Mitchell [34], which work by repeatedly dividing the histogram of the data into blocks, depend solely on the use of the RGB colour space for their algorithms. Given that, in most natural scenes, the colours are generally unsaturated [26], then the correlation between the red, green and blue components will be high. The effect of a linear transformation to a more uncorrelated set of primaries may itself be sufficient to quantise the image in some cases. (It should be mentioned that in the field of computer graphics, images often contain artificially saturated colours.)

2.2. Colour Space Transformations

The aim of this study is to quantise the image in such a way as to provide the minimum distortion in the resulting image from a perceptual viewpoint. The interest is not in finding, on a pixel-by-pixel basis, the smallest difference between the input primaries - red, green and blue - and the output. Consider the diagram shown in figure 2.1. It is a plot of some experimental results obtained by MacAdam (from Wysecki and Stiles [35]), which show the ellipses of just noticeable differences about some 25 different chromaticities, plotted on the CIE (*x*-*y*) chromaticity chart [9]. The circumferences of each ellipse indicate the distance travelled on the chart from the ellipse center before the observer noticed, on average, a definite change in the displayed colour. It is apparent that human perception of chromaticity change is not linearly related to the CIE co-ordinate system. Since the CIE XYZ colour space is linearly related to the RGB colour space in which the display hardware operates, attempting to minimise the differences between input and output of RGB triplets is not guaranteed to provide the closest colour match as far as the average observer is concerned. The diagram in figure 2.2 (from Durrett [36]) shows these same ellipses plotted on the CIE-UCS (*u*-*v*) chart [15], a linear modification of the *x*-*y* chart. There is a visible improvement, but the radii of the ellipses still vary considerably. There is considerable ongoing research aimed at providing a linear transformation of the x-y chart which aproximates circles rather than ellipses (although exactly equal-radii circles cannot be generated this way [13]). The importance of the linear requirement is that without it, the centre-of-gravity principle (which defines the chromaticity of a two colour mixture as the luminance-weighted average of the two chromaticities) would be lost, and colour mixtures within the space would become unpredictable [26].

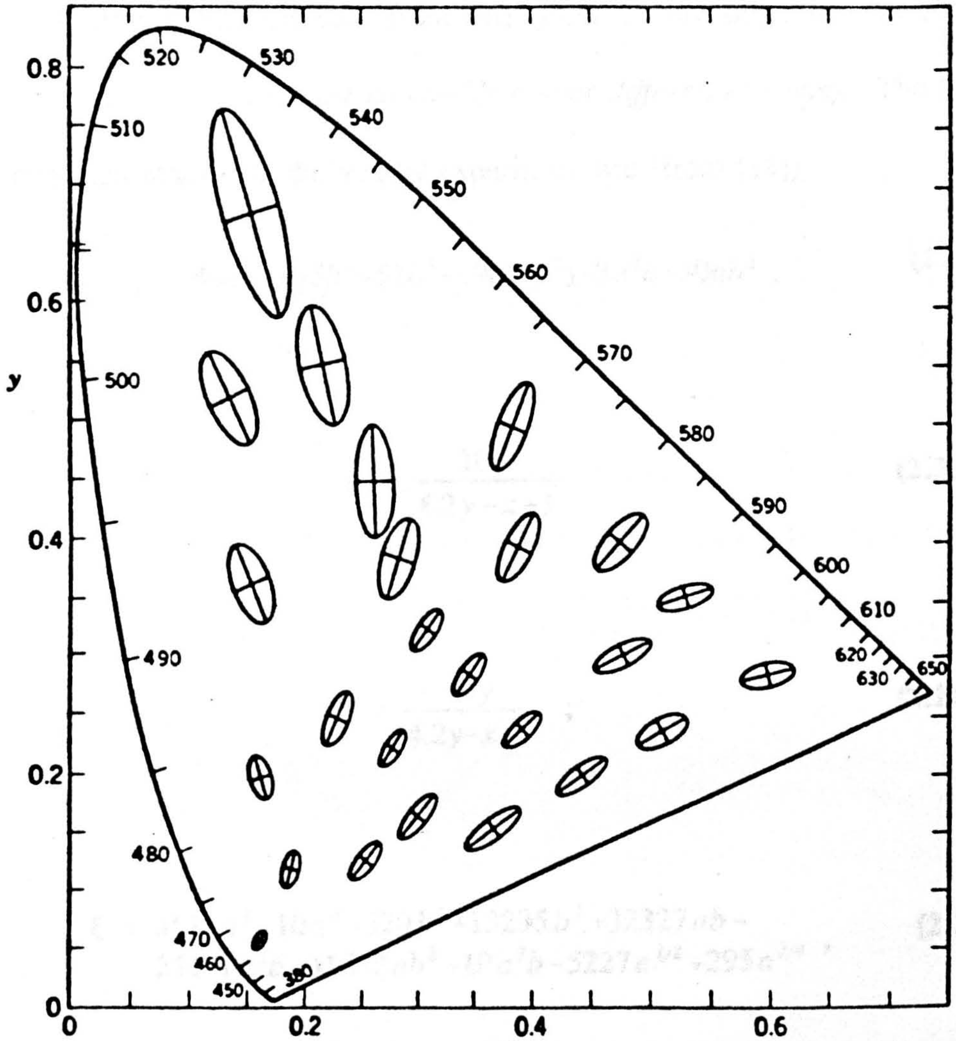


Figure 2.1. MacAdam's ellipses plotted on CIE x-y chromaticity chart. (Source: Wysecki and Stiles [35]).

2.3. MacAdam's Geodesic

MacAdam [14], allowing for the loss of centre-of-gravity, has generated a pair of non-linear transforms which generate the almost circular ellipses shown in figure 2.3 (from MacAdam [14]). The resulting chart is called a geodesic, because the straight-line distance from one point to any other on the chart involves the least number of *just noticeable* colour *differences* (jnds). The two equations which MacAdam derived by experiment are (from [14])

$$\eta = 404b-185b^2+52b^3+69(1-b^2)-3a^2b+30ab^3 \,, \qquad (2.1)$$

where

$$a = \frac{10x}{4.2y-x+1} \qquad (2.2a)$$

and

$$b = \frac{10y}{4.2y-x+1} \;; \qquad (2.2b)$$

$$\xi = 3571a^2-10a^4-520b^2+13295b^3+32327ab- \\ 25491a^2b-41672ab^2+10a^3b-5227a^{1/2}+295a^{1/4} \,, \qquad (2.3)$$

where

$$a = \frac{10x}{2.4x+34y+1} \qquad (2.4a)$$

and

$$b = \frac{10y}{2.4x + 34y + 1} \quad . \qquad (2.4b)$$

Of course, the symbols x and y denote the CIE chromaticity co-ordinates. Figure 2.4 (from MacAdam [14]) illustrates the relationship between the CIE and MacAdam domains for loci of x and y.

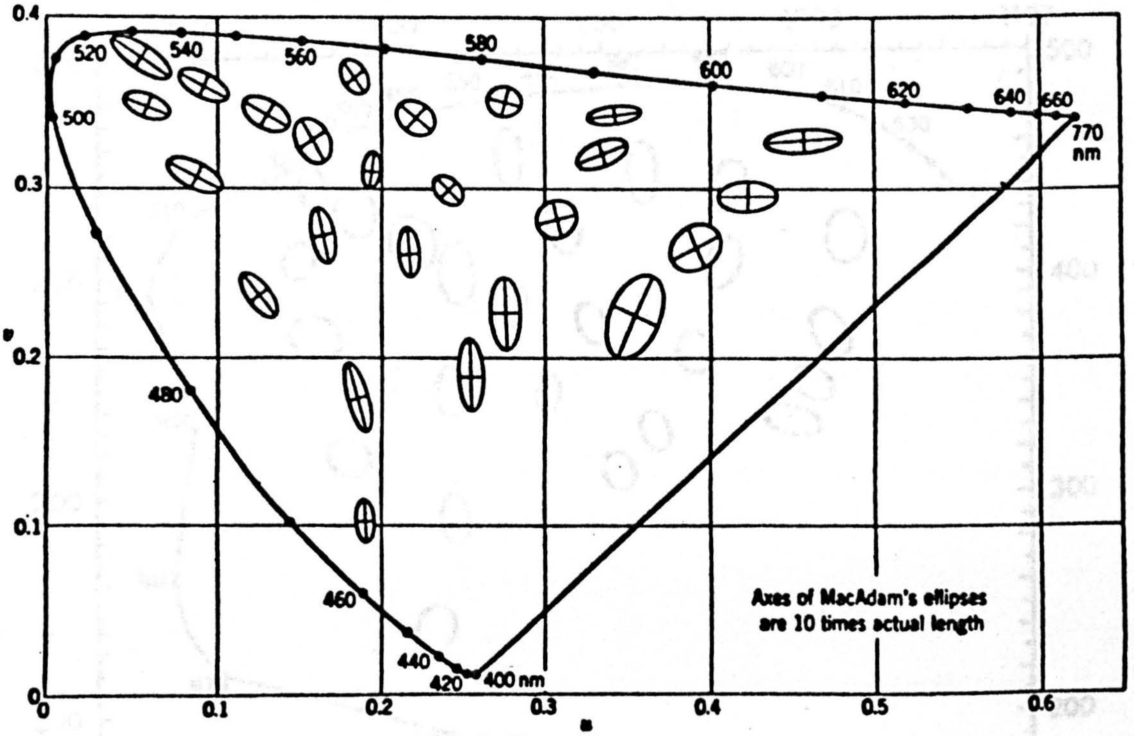


Figure 2.2. MacAdam's ellipses plotted on CIE-UCS chart.

(Source: Durrett [36])

Apparently, the complex, non-linear transformation described above is not algebraically reversible. Such a restriction is a serious impediment to a quantisation scheme, since it is a requirement that the output data are RGB triplets, to be sent to the colour guns of the display. A method of transforming between (*x*,*y*) and MacAdam chromaticity spaces using a lookup table (LUT) technique will be described subsequently, but consider first the effect of linear transformations alone on the chromatic content of a colour image.

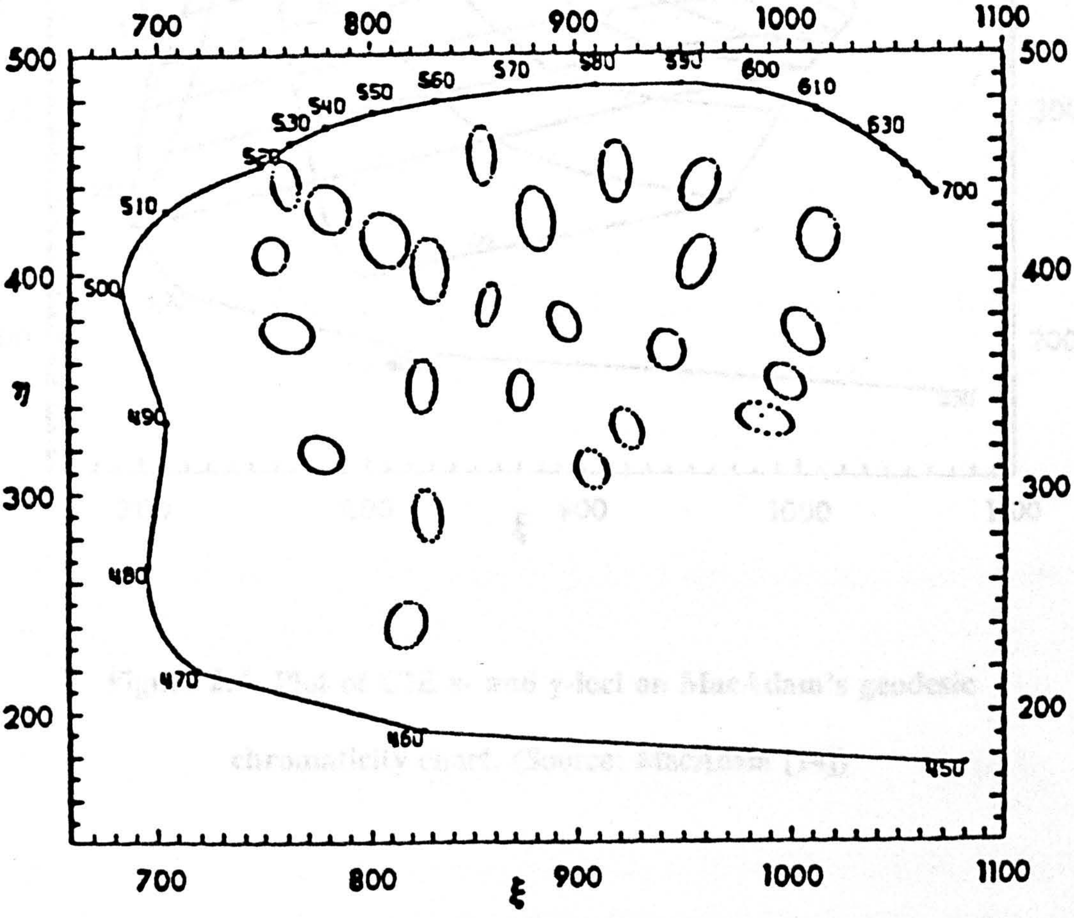


Figure 2.3. MacAdam's ellipses plotted on geodesic chromaticity chart. (Source: MacAdam [14])

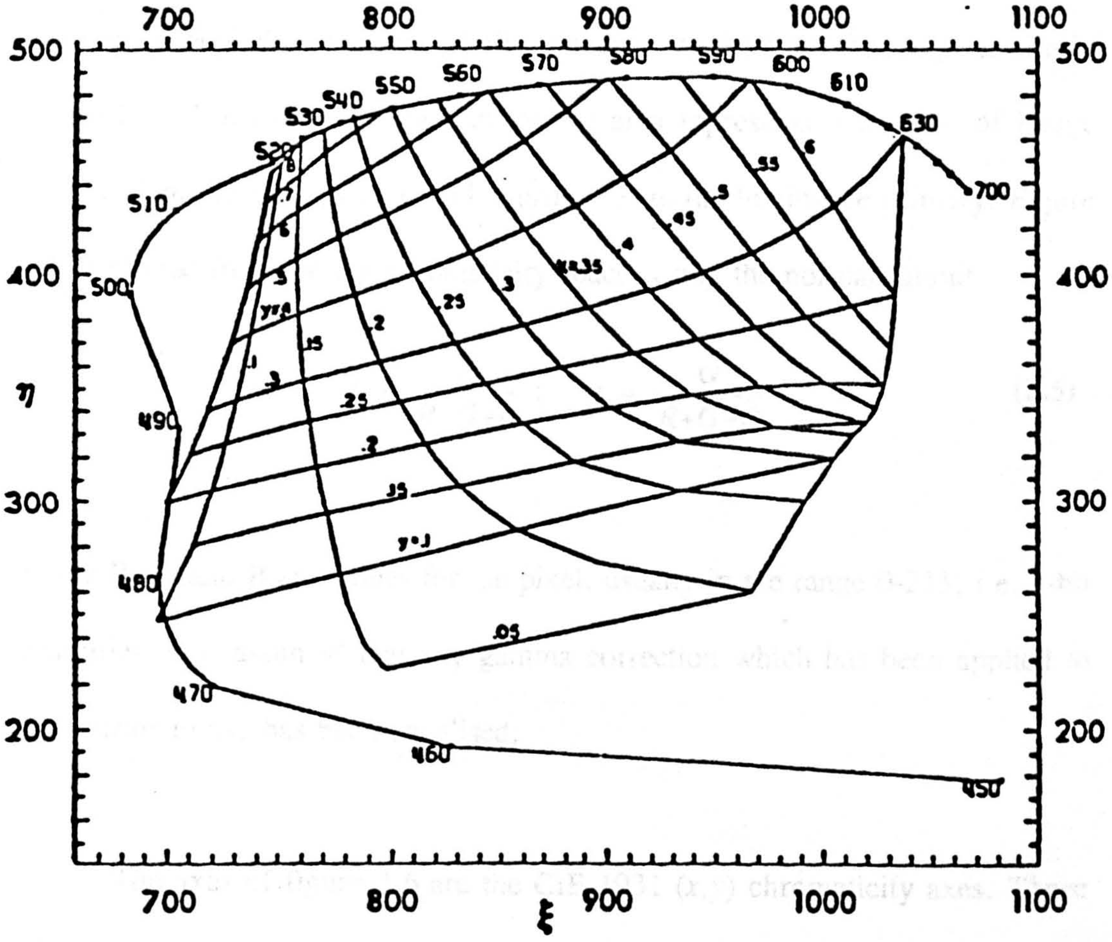


Figure 2.4. Plot of CIE x- and y-loci on MacAdam's geodesic chromaticity chart. (Source: MacAdam [14])

Figures 2.5, 2.6, 2.7 and 2.8 indicate the extent of some chromaticity diagrams occupied by the data (at all luminance levels) from the image shown in figure 2.9. The change in the size of the area represents a transfer of image energy from the red, green and blue primaries to the luminance primary. Figure 2.5 is plotted from the r-g chromaticity space, using the normalisations

$$r = \frac{R}{R+G+B} \; ; \quad g = \frac{G}{R+G+B} \tag{2.5}$$

where **R**, **G** and **B** are values for the pixel, usually in the range 0-255, i.e. 8-bit quantities. It is assumed that any gamma correction which has been applied to these prior to use has been equalised.

The axes of figure 2.6 are the CIE 1931 (x,y) chromaticity axes. Those of figure 2.7 are the CIE 1960 UCS (u,v) space, related to the (x,y) space by the following transforms (from [15]):

$$u = \frac{4x}{-2x+12y+3} \; ; \quad v = \frac{6y}{-2x+12y+3} \tag{2.6}$$

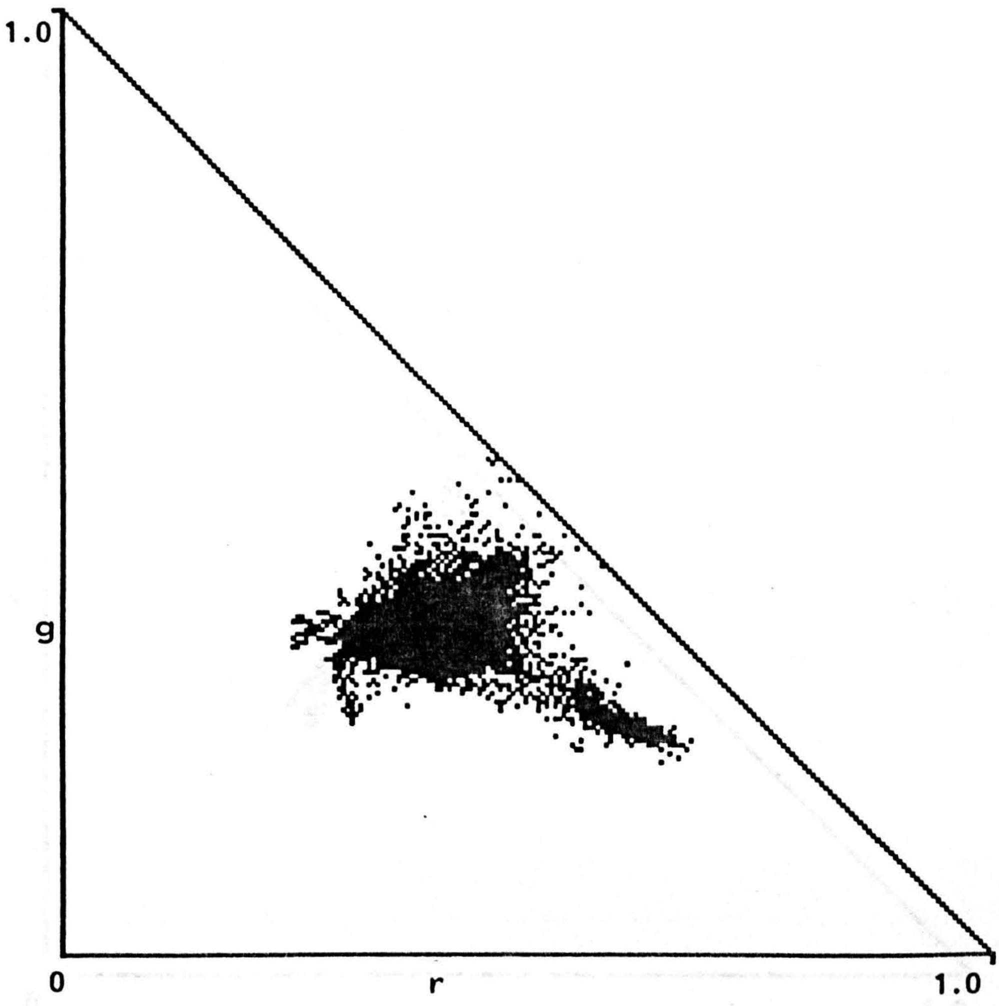


Figure 2.5. Chromaticity locus of colour image in figure 2.9 plotted on r-g chromaticity chart.

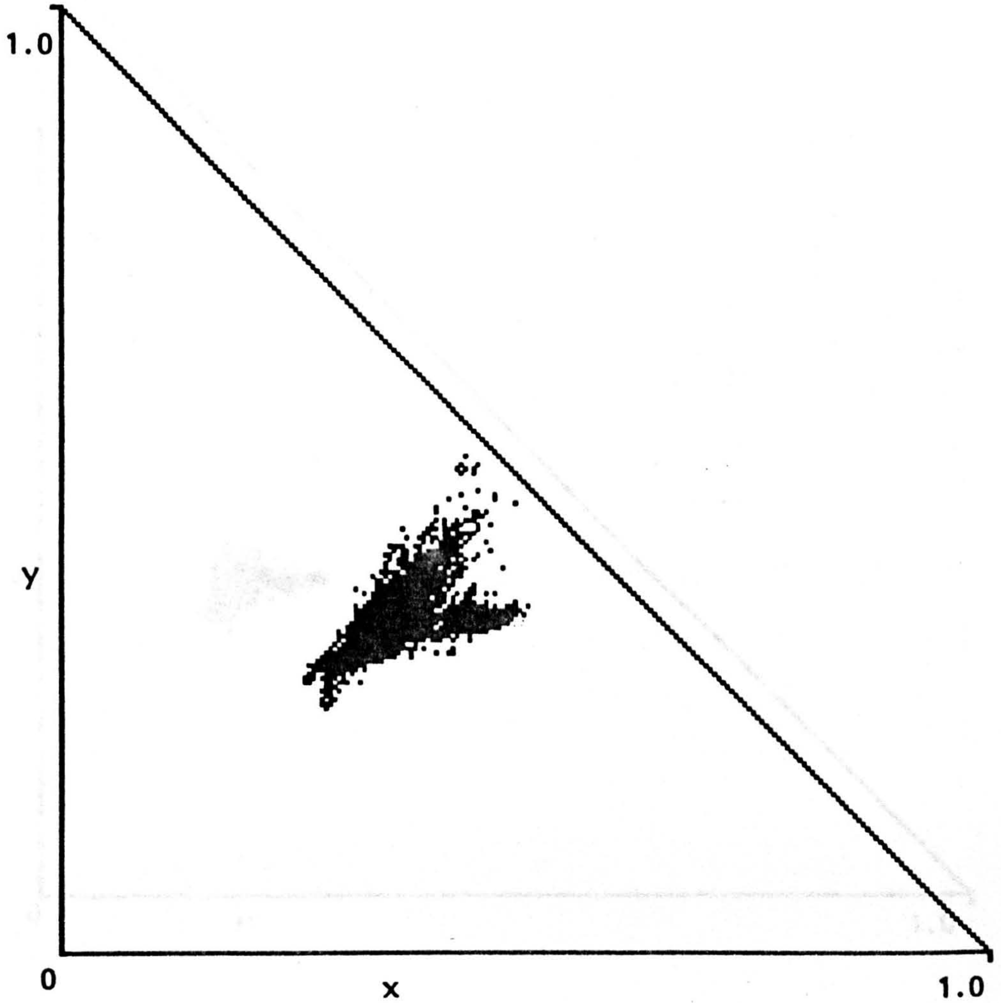


Figure 2.6. Chromaticity locus of figure 2.9 plotted on CIE x-y chromaticity chart.

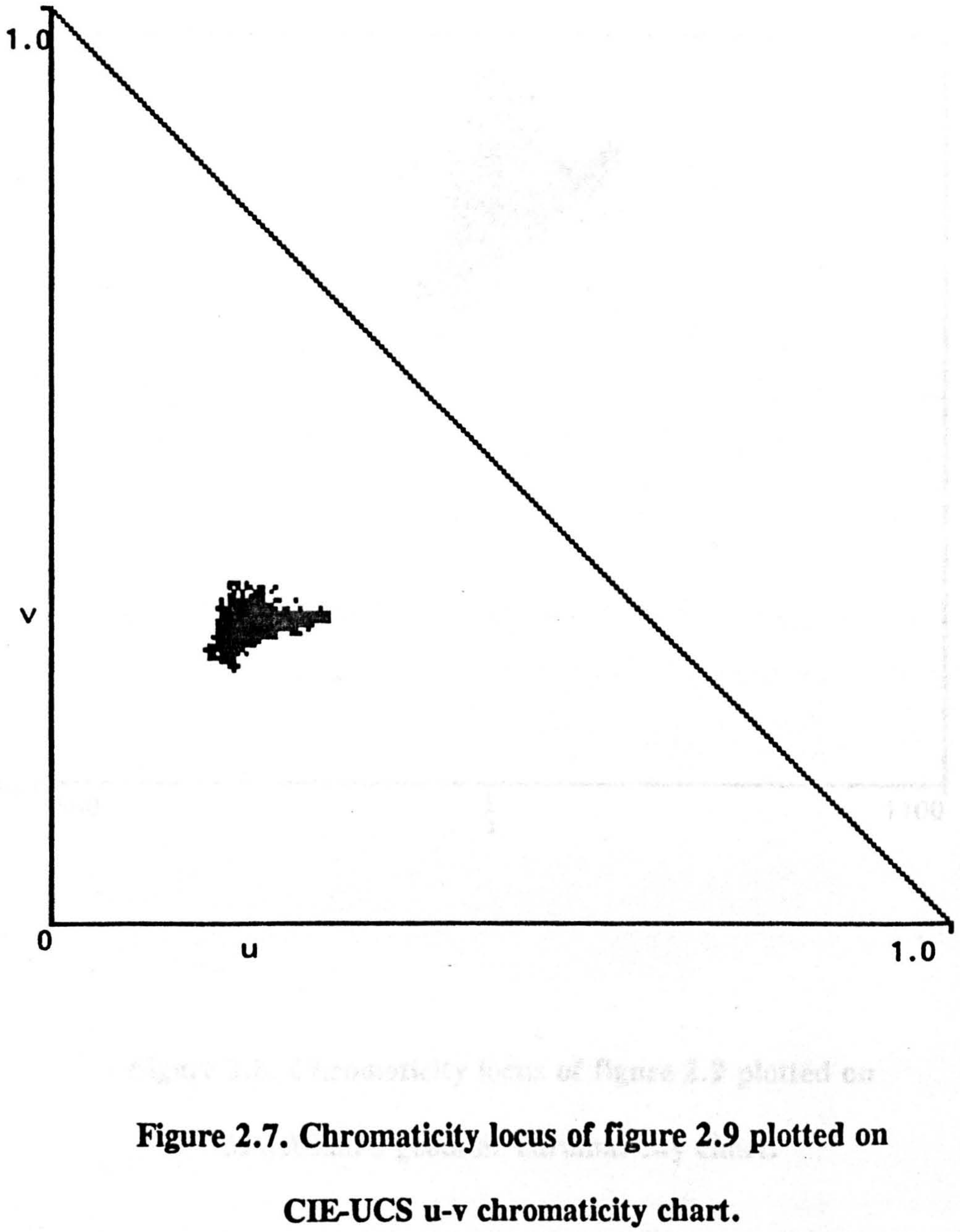


Figure 2.7. Chromaticity locus of figure 2.9 plotted on CIE-UCS u-v chromaticity chart.

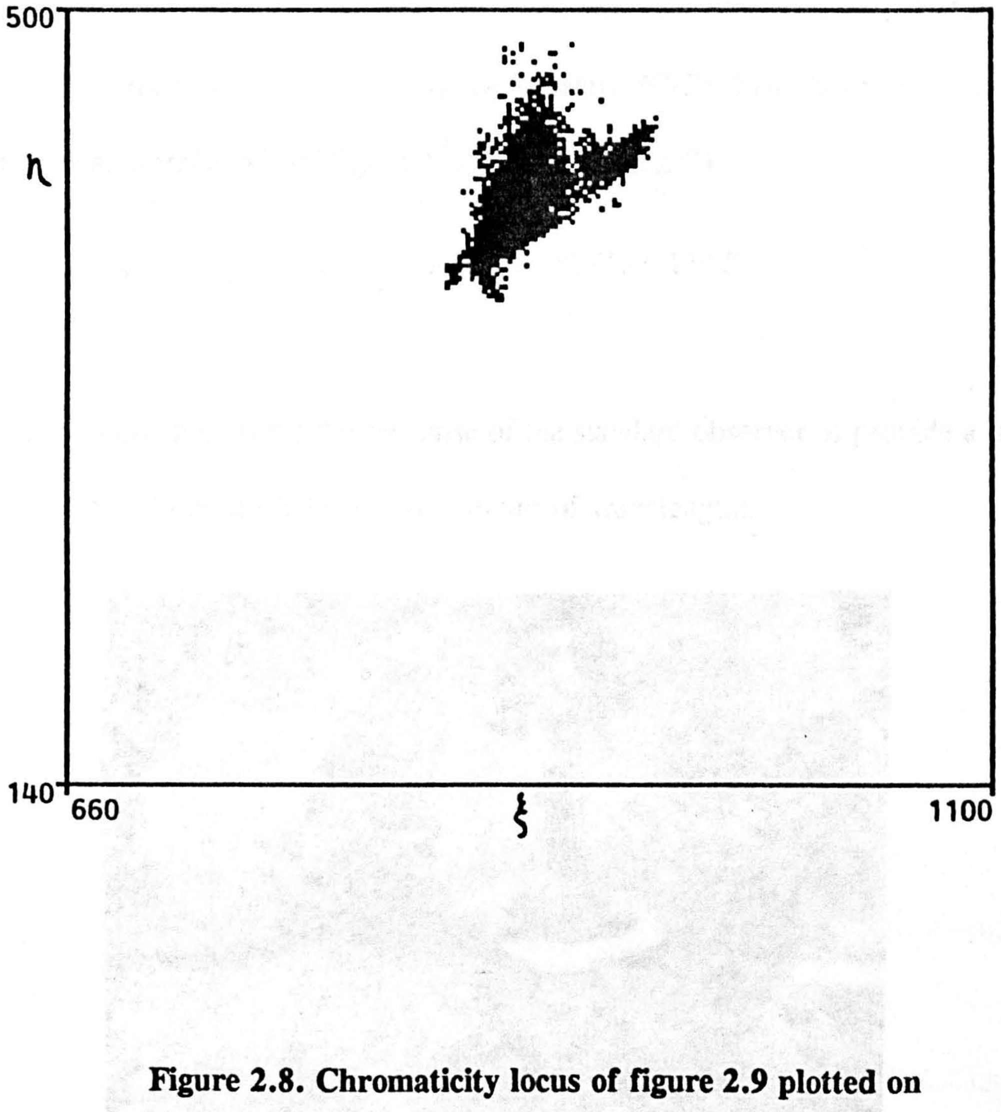


Figure 2.8. Chromaticity locus of figure 2.9 plotted on MacAdam's geodesic chromaticity chart.

Figure 2.8 is a plot in the MacAdam geodesic. In all of the transformations, the lightness co-ordinate, which will be referred to from now on as the grey-level, or the intensity, or the luminance, of the colour, is related to the red, green and blue values by the standard (CIE) formula (normalised to a particular distribution of light known as illuminant C)

$$Y = 0.299\boldsymbol{R} + 0.587\boldsymbol{G} + 0.114\boldsymbol{B} \tag{2.7}$$

This formula represents the response of the standard observer to provide a match to an equal-intensity light of any mixture of wavelengths.

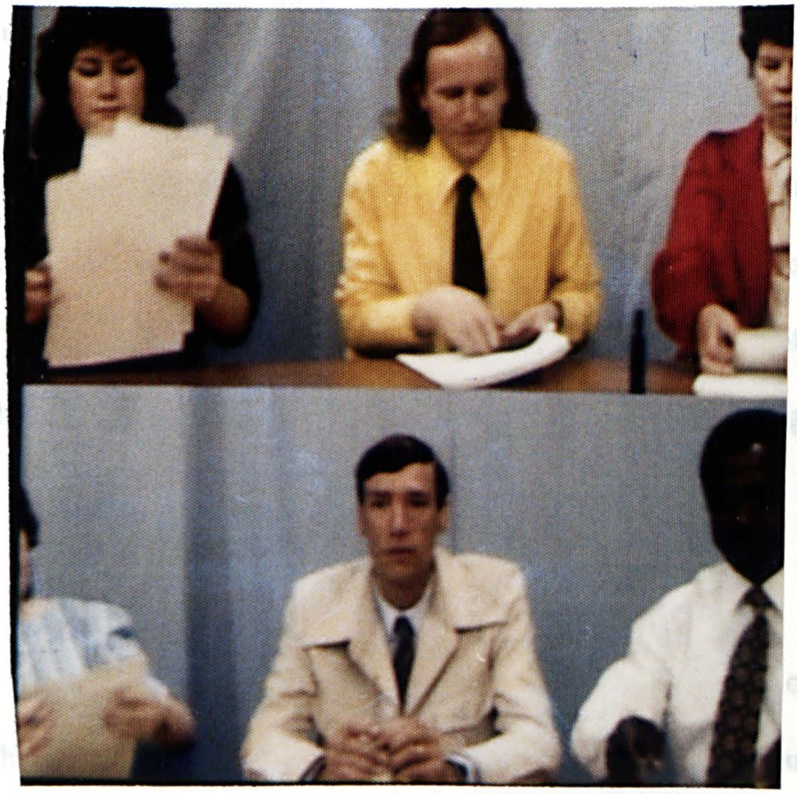


Figure 2.9. Colour image used for chromaticity plots in figures 2.5, 2.6, 2.7 and 2.8.

2.4. Quantisation Parameters

From the previous figures, then, it is apparent that simply transforming from RGB to one of the chromaticity-luminance spaces seems to provide a considerable compression of the "colour" in the image. At this point, some consideration should be given as to how coarse a quantisation, i.e. how many bits-per-pixel, is likely to be acceptable. It is widely held, for instance, that the untrained eye is only capable of discriminating about 64 grey shades, or 6 bits. What about chromaticity ? Consider figure 2.8 and Maxwell's geodesic once again. The ellipses in the geodesic, which indicate typical ability to perceive colour difference, indicate that about 10 units in the geodesic correspond to a jnd. Hence, under the assumption that one can add these across the chart, figure 2.8 shows that about 4 bits are needed to adequately represent the typical image. But to provide a finer quantisation scheme, consider figure 2.10. (from Fink [37]). This figure shows the (x,y) chart with the typical gamut or extent of realisable colours contained inside the triangle. Notice that the true range of colours lies within the triangle with sides defined approximately by the following inequalities:

$$0.14 \leq x \leq 0.66 \ ; \quad 0.07 \leq y \leq 0.72 \ ; \quad y \geq 0.5x \ ; \quad x+y \leq 1.0 \qquad \textbf{(2.8)}$$

Plotting this triangle on the geodesic generates the relationship shown in figure 2.11. The range of realisable colours is reduced to an area bounded by the following inequalities:

$$720 \le \xi \le 1040 \;; \qquad 240 \le \eta \le 480 \tag{2.9}$$

Note that MacAdam's experimental results indicated a threshold of at least 10 units on his chart before a colour change was detected on average. With this information, it should be reasonable to halve the resolution of the space as a quantisation, without loss of information. This will be discussed further when the design for the LUT is presented.

Returning to the discussion of perceived distortion, it is apparent that MacAdam's chart is a metric space, because straight line vectors correspond to the smallest amount of perceived equiluminous colour change. The nature of this space allows an analytic measure of colour distortion, in that the distance between any two points on the chart is linearly related to the perceived difference, a feature known as *proportionality*, and the perceived difference is independent of the absolute positions of the points, a feature known as *homogeneity*. In fact, it is now meaningful for us to apply a simple measure like the Mean Square Error (MSE) measure,

$$MSE(x,\hat{x}) = \frac{1}{N}\sum_{i=0}^{N-1}(x_i-\hat{x}_i)^2 \,, \tag{2.10}$$

to the quantised pixel chromaticity values, and thus provide a reasonable account of the average distortion perceived by the viewer. At this point, it should be mentioned that the role of the quantised grey-levels in this distortion measure has not been explored. This will be discussed in future sections.

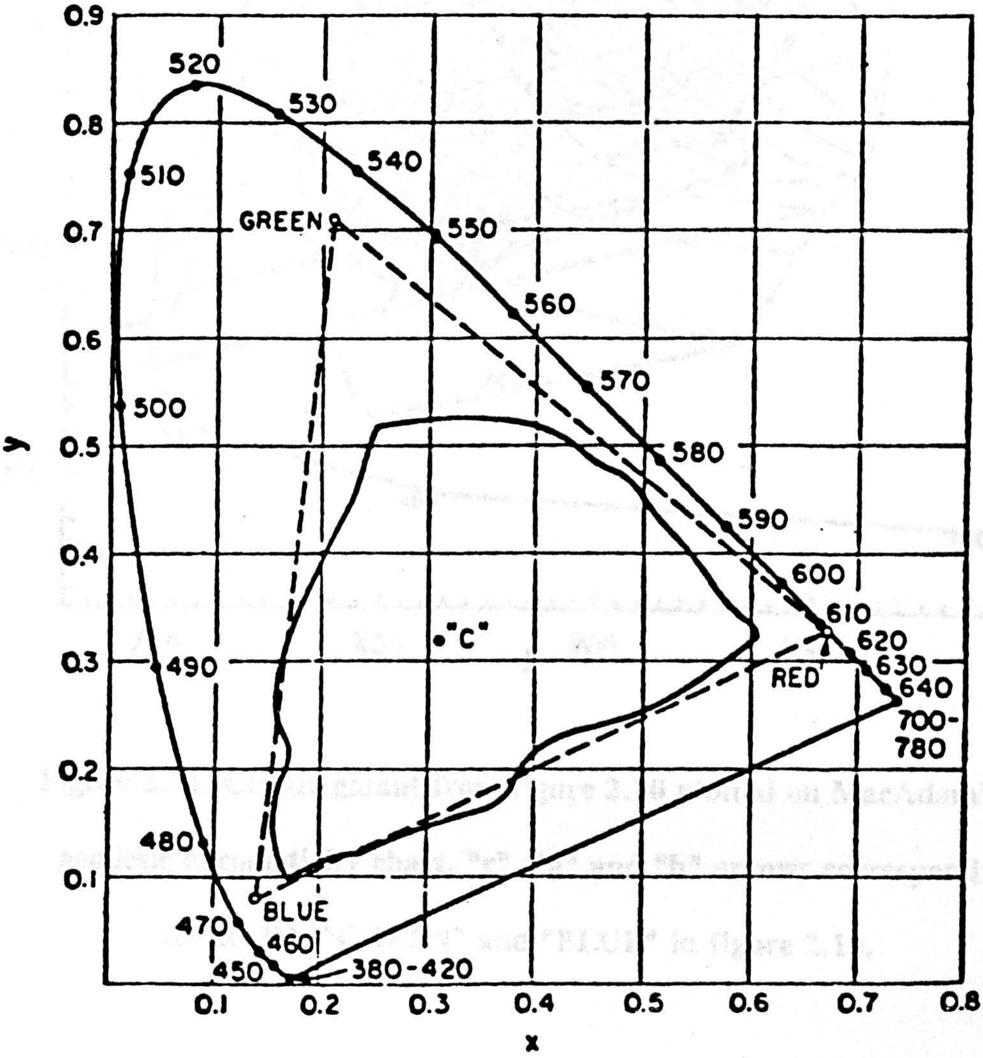


Figure 2.10. Typical colour gamut (Solid) and r-g chromaticity (Dotted) plotted on CIE x-y chromaticity chart. Point "C" corresponds to chromaticity of illuminant C. (Source: Fink [37])

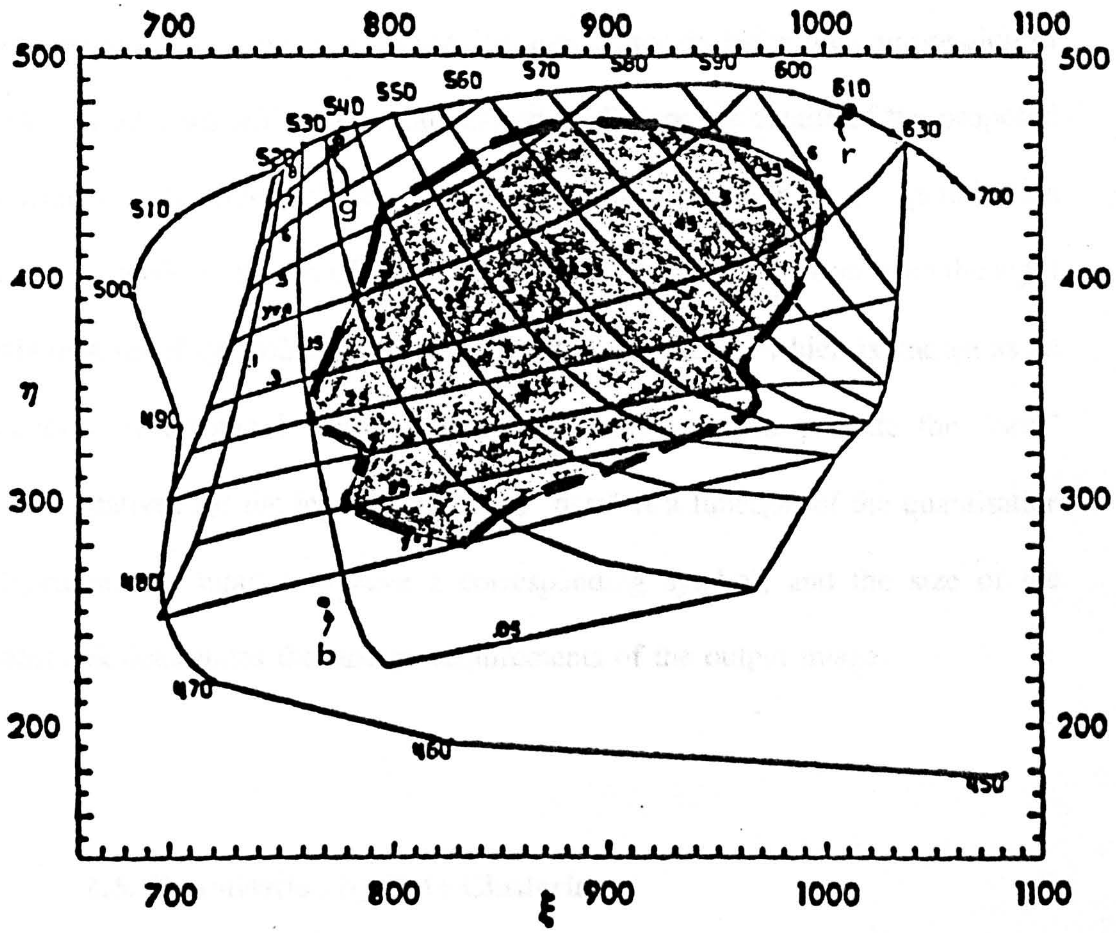


Figure 2.11. Colour gamut from figure 2.10 plotted on MacAdam's geodesic chromaticity chart. "r", "g" and "b" arrows correspond to "RED", "GREEN" and "BLUE" in figure 2.10.

Having considered the various colour spaces in which perceived distortion closely matches distance through the space, and the use of measurements of such distortion, the discussion turns to the implementation of an image quantisation scheme. The goal, which was somewhat arbitrarily chosen, is to reproduce 24 bpp images at around 4 bpp, with minimal perceived colour distortion. This quantisation scheme would thus provide approximately twice the compression of common schemes with less visible distortion. Before the details of the proposed scheme are described, those elements which are common to all quantisation schemes should be presented. The procedure involves the selection from the input data of a set of *symbols* or *codewords*, vectors or scalars, which is known as an *alphabet* or *codebook*; these symbols are considered to provide the "best" representatives for the input data, where "best" is a function of the quantisation algorithm. All input data have a corresponding symbol, and the size of the codebook determines the storage requirements of the output image.

2.5. Quantisation by Data Clustering

The particular quantisation scheme investigated in this study makes use of a *clustering* algorithm, rather than choosing a series of fixed quantisation levels and a series of related grids. The reasons for choosing a clustering method rather than fixed levels are:

1. The content of typical images such as are encountered in videoconferencing contain few large "groups" of colours, i.e. a histogram of the image in *RGB* space would contain a few strong peaks or *modes*.

2. Using fixed quantisation levels on such an image would be a waste of bits: few symbols would be produced from the available alphabet. Note that no entropy coding is being considered for this quantisation scheme.

2.6. The "k-means" Test

The clustering algorithm involves the execution of an iterative scheme designed to "relax" to a stable solution to the problem of finding a fixed number of representative chromaticities and grey-levels. One of the most well-known is the *k-means test*, developed by MacQueen [38]. The algorithm is a simple one and has been proven to converge, although it is noted that the final solution does depend on the choice of initial guesses, i.e. the algorithm is not guaranteed to find a global minimum or maximum. The k-means test resembles a vector quantiser, such as that proposed by Linde *et al.* [23] In both these algorithms, the relaxation involves iteratively computing, in an arbitrary n-dimensional Euclidean space, the *minimum distortion vector* in the space. Linde *et al.* christened this vector the *centroid,* or *centre of gravity,* of a part of the signal space. The ***i-th*** centroid of the space, c_i is defined as

$$c_i = \sum_{j=0}^{N-1} \frac{x_j p(x_j)}{\sum\limits_{k=0}^{N-1} p(x_k)}, \tag{2.11}$$

where x_j is the ***j-th*** codeword which lies in the ***i-th*** sub-space, and ***p()*** is the probability of occurrence of that codeword. In the case of an unknown distribution, this value is usually the number of occurrences of the codeword.

Once the number of symbols in the alphabet has been decided, and a histogram or some other measure of the sample signal distribution has been determined, the k-means algorithm proceeds in the following manner:

1. Assume that the signal space is divided into k sets S_i, with local centroids c_i, and $i=0,1,...k-1$ i.e. k symbols comprise the alphabet.
2. Assume that there are N input vectors, or symbols x_j, $j=0,1,..,N-1$ in the signal.
3. Repeat until there is no change in the value of any centroid after iteration n :

For each vector x_j, $j=0,1,..,N-1$,

$$x_j \;\varepsilon\; S_i \;\mathit{if}\; |x_i-c_i^{(n-1)}|<|x_j-c_l^{(n-1)}|,\; 0\le l<k,\; l\ne i\ . \qquad (2.12)$$

For each centroid $c_i^{(n-1)}$, $i=0,1,..,k-1$,

$$c_i^{(n)} = \sum_{j\varepsilon S_i} \frac{x_j p(x_j)}{\sum\limits_{k\varepsilon S_i} p(x_k)}\ . \qquad (2.13)$$

4. Retain the final set boundaries S_i, $i=0,1,\ldots,k-1$ as the assigned sub-spaces for each symbol in the alphabet.

2.7. Quantiser Colour Space Conversions

The signals which are supplied to the quantiser in this scheme are the luminance values for each pixel, and the chromaticity values in the MacAdam space. A series of transformations are required to move from the input RGB signal space to the quantiser inputs. The MacAdam chart is a transformation of the CIE colour space, so the data must first be transformed from RGB to XYZ triplets. This is achieved simply by matrix multiplication (see for example Pratt [39] for an extensive list of linear transformations between colour spaces.) The form is as follows:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.607 & 0.174 & 0.201 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.117 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (2.14)$$

and the XYZ triplet is in turn rendered as luminance (Y) and chromaticities (x,y) by equations similar to those in (2.5), i.e.

$$x = \frac{X}{X+Y+Z} \; ; \quad y = \frac{Y}{X+Y+Z} \qquad (2.15)$$

Now consider the conversion between the CIE and MacAdam chromaticity spaces. It is impossible to provide an analytic method, so we must seek other means. A simple way is to store tables of forward and inverse transformations, and link the two to form an integrated look-up table, or *LUT.* Referring to the discussion of the geometric formulae which specify the extent of the CIE and MacAdam spaces, we note that with a halving of the resolution in the MacAdam domain, the numerical distances are in the range of 0 to 255. Quantisation of chromaticity values to 8-bit values, therefore, is sufficient to provide the range required. The error introduced by this rounding process will be discussed in a later section.

The number of indices required for the forward transformation can be derived from the inequalities (2.8) and the 8-bit quantisation per chromaticity ordinate. The area of the triangle in figure 2.10 can be approximated by a rectangle and two triangles as

$$\begin{array}{l}(0.72-0.33)\cdot(0.28-0.14)\ + \\ \left(\dfrac{(0.72-0.33)\cdot(0.66-0.28)}{2}\right) + \\ \left(\dfrac{(0.33-0.07)\cdot(0.66-0.14)}{2}\right)\end{array} \approx 0.20 \qquad (2.16)$$

and if this number is multiplied by the square of 256 the approximate number of indices required is

$$(0.2)\cdot(256)\cdot(256) \approx 13110\ \textit{indices}\ . \qquad (2.17)$$

To implement the table in hardware, an additional $(0.72-0.07)\cdot 256=167$ entries are needed, to indicate the number of indices in each row of the table.

The upper limit on the number of entries required for the reverse transformation, from the inequalities in Eq. (2.9) (at half resolution) with integer accuracy is

$$(520-360)\cdot(240-120) \quad = \quad 19200 \textit{ indices} \; . \qquad (2.18)$$

which means that the two-way LUT will have approximately 32500 entries, with each entry containing two 8-bit numbers. The table therefore requires about 64 kilobytes of storage. Note that because the "origins" of both the transform spaces are not (0,0) a subtraction and addition will be required along with each lookup operation.

Statistics gathered from a program which generates this LUT show that the rounding of the data occasionally results in more than one entry per index, so an averaging process must be performed to provide a reasonable result from the look-up operation. The maximum number of entries per index was found to be seven, with an average of 1.4, and the maximum standard deviation in the averaging process was 1.667 (using 8-bit values), less than one percent of the maximum value along either axis.

Given that the centre-of-gravity principle is not preserved in the MacAdam chromaticity space, it is important to consider the propriety of cluster analysis and LUT cell-averaging given such conditions. In the introduction, the centre-of-gravity principle was noted as being essential if one wishes to predict the chromaticity of a mixture of colours. The equations can be found in, for example, Limb *et al.* [26] However, in this study the interest is not in colour mixtures, but rather in colour distortions or differences. The clustering algorithm is designed to find those sub-spaces within which all vectors are perceived as being closer to a particular representative vector than to any of the other representatives; the result of their mixture is not considered. Averaging entries in the LUT is actually an averaging of vectors in the (x,y) space which are mapped to the same "box" in the MacAdam space; the calculated average vector is also a location in the (x,y) space, and it may be envisaged as being mapped from the "centre" of the box.

2.8. Quantisation of Achromatic Data

Turning the discussion to the quantisation of the achromatic part of the image signal, the interest once again is in the perceived distortion in the output. Studies of the visual system indicate (See for example Hecht [40], Cornsweet [41], Michelson [42]) that the visual system has a linear response to the *contrast* in an image rather than to the absolute values of the pixels. Models of the visual system using an analogy to film development (Stockham [31], Hunt [43]) hold that the visual system is linearly sensitive to the natural logarithm of the signal. (Some others, such as Mannos [44], Stevens [45] and Pearson [46], argue that the base for the power law is not exactly *e*. Mannos computes from experiment a number closer to 0.33 than to 0.27. Pearson proposes 0.33 for TV images and higher values when the surround is very bright, such as hardcopy images on white paper.) For videoconferencing applications, using a fixed quantisation, then, the scale should be approximately log-linear, as employed by Kurz [17], for example. The clustering algorithm, however, behaves poorly when the achromatic values are converted to logarithms. The tendency is for all the pixels to be associated with one or two centroids only. Because of this empirical drawback, and to avoid further use of floating-point arithmetic, the absolute luminance values were used within the cluster algorithm throughout this study. This course of action undoubtedly seems to fly in the face of the established literature, but the improved results obtained from using absolute data suggested that this approach was the better of the two. This subject will be discussed further at the end of the chapter, and in the concluding chapter.

2.9. Quantiser Implementation Details

Now that the colour space-conversions and the clustering algorithm have been investigated, consider the next part of the quantisation process, which involves the quantised luminance and chromaticity symbols and their corresponding absolute values. The clustering itself takes place in the luminance and MacAdam colour spaces. Because these signals are quantised independently, it is possible to generate quantised vectors (Y,η,ξ) which in fact lie slightly outside the RGB colour space. In other words, the regenerated colours may be too saturated, lying outside the triangle shown in figure 2.9. This is a drawback which must be addressed in a manner which provides both a minimum perceptual distortion and a quantised colour which is also found in the generated codebook. It is assumed that the codebook which will be searched contains equiluminous chromaticities. In fact, as long as separate codebooks are found for each luminance sub-set, then the chromaticities will be nearly equiluminous, and the condition is thus satisfied. The procedure is straightforward. The codebook is searched until the closest chromaticity to the chosen one (by its Euclidean distance, on the MacAdam chart) which is realisable in RGB space, is the new choice.

Once the quantised luminance and MacAdam chromaticity values have been chosen from the codebooks for each pixel, then the (x-y) chromaticities are obtained from the LUT and the XYZ values may be found simply from the following:

$$(X+Y+Z)_q = \frac{Y_q}{y_q} \; ; \qquad (2.19a)$$

$$X_q = x_q \cdot (X+Y+Z)_q \; ; \qquad (2.19b)$$

and

$$Z_q = (1-x_q-y_q) \cdot (X+Y+Z)_q \qquad (2.19c)$$

where the subscript *q* indicates a value which has been quantised. From this point, the quantised RGB values can be found by matrix multiplication (see Pratt [39]). A flow chart of the complete quantisation procedure, excluding the generation of the LUT, is shown in figure 2.12.

2.10. Sources of Quantiser Error

Now that the complete quantisation scheme has been described, consider the various sources of numerical error that contribute to distortions in the processed image. If the individual stages of the scheme are scrutinised, the main sources of error are seen to be those where real arithmetic is needed, leading to rounding and truncation errors. These occur at the following transformations :

$$\begin{array}{l} RGB \rightarrow XYZ \; ; \quad XYZ \rightarrow Yxy \; ; \quad xy \rightarrow \xi\eta \rightarrow \frac{\xi}{2}\frac{\eta}{2} \; ; \\ \frac{\xi}{2}\frac{\eta}{2} \rightarrow xy \; ; \quad Yxy \rightarrow XYZ \; ; \quad XYZ \rightarrow RGB \; . \end{array} \qquad (2.20)$$

The initial matrix multiplication, which was modified for integer arithmetic by multiplying all the coefficients of the matrix by 4096 and right-shifting the result twelve times, introduces a maximum error in any row of 1.81×10^{-4}. For the maximum signal, (255,255,255), the error is therefore less than 0.04, so the integer result may vary by at most 1. With the inverse matrix multiplication, the maximum error is 1.81×10^{-4}; the error may be at most 0.05, so the true value may differ by at most 1.

Rounding the (x,y) chromaticity values to the nearest integer produces a maximum error of 0.0039. The maximum error transferred to the MacAdam space (with halved resolution) is 1 in each ordinate. This figure includes any rounding in the MacAdam space. The error introduced by the transformations in the LUT are, on average, at most 1 after both a forward and reverse transform. Converting integral chromaticity values back to real numbers again introduces a maximum error of 0.0039.

There will also be numerical errors introduced into the clustering process, because after each iteration, the centroids are set as integers. The effect of these errors on the overall distortion in the image will be variable in character.

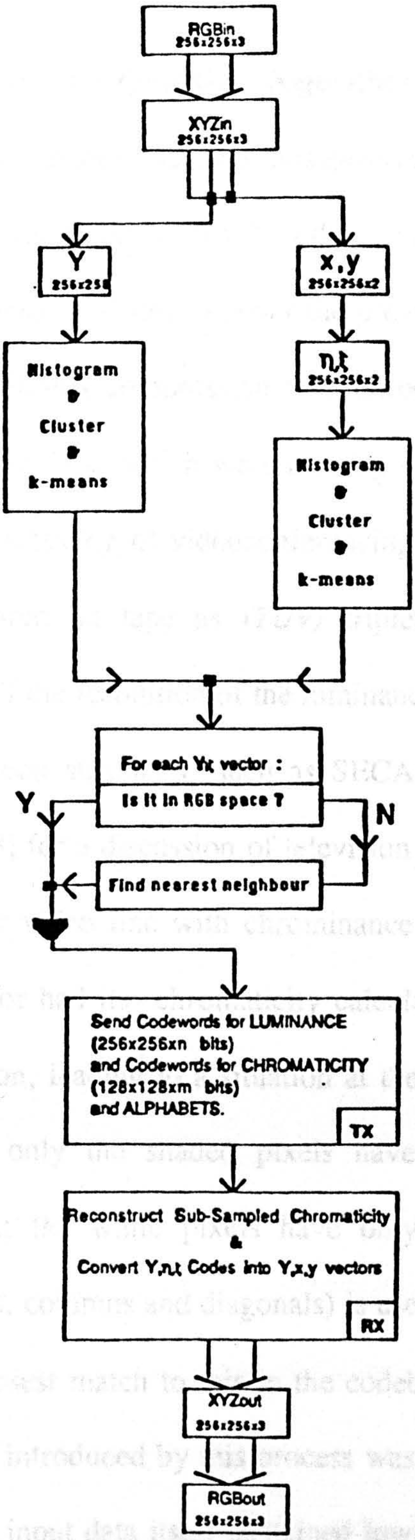


Figure 2.12. Quantisation flow diagram featuring Transmitter (Tx) and Receiver (Rx) Elements.

2.11. Results from The Quantiser Algorithm

The investigation into the efficacy of this quantisation scheme involved the implementation of three variations on the flow diagram shown in figure 2.12, in order to improve the perceived distortion over the previous variation. As well as these variations, an additional compression was introduced into each process, motivated by the available data, which were a series of digitised frames from a videotape suitable for the testing of videoconferencing compression algorithms. The data in fact appeared on tape as *(YUV)* triplets, but the chrominance information was only half the resolution of the luminance. Because of this factor, and also because television standards, such as SECAM (see Wright [47] and Carnt and Townsend [48] for a discussion of television standards), also typically furnish only every other video line with chrominance information, only every fourth input *(RGB)* vector had its chromaticity calculated by the "transmitter" section of the quantisation, leading to a situation at the "receiver" as illustrated in figure 2.13, where only the shaded pixels have both a luminance and chromaticity component; the white pixels have only the luminance. Linear interpolation (along rows, columns and diagonals) is used to provide a best guess chromaticity, and the closest match to this in the codebook is used in its place. The additional distortion introduced by this process was not noticeable, although again it is noted that the input data itself contained low-resolution chrominance. The algorithm itself does not introduce any further perceived distortion, then, by halving the resolution at the quantisation stage. Note that the full resolution of the input image was used in the quantisation process, before discarding the extra chromaticity values. Using a luminance codebook with sixteen values, then, and a chromaticity codebook with eight values, the output bit-rate for this quantisation is 4 + (3/4), or 4.75 bpp.

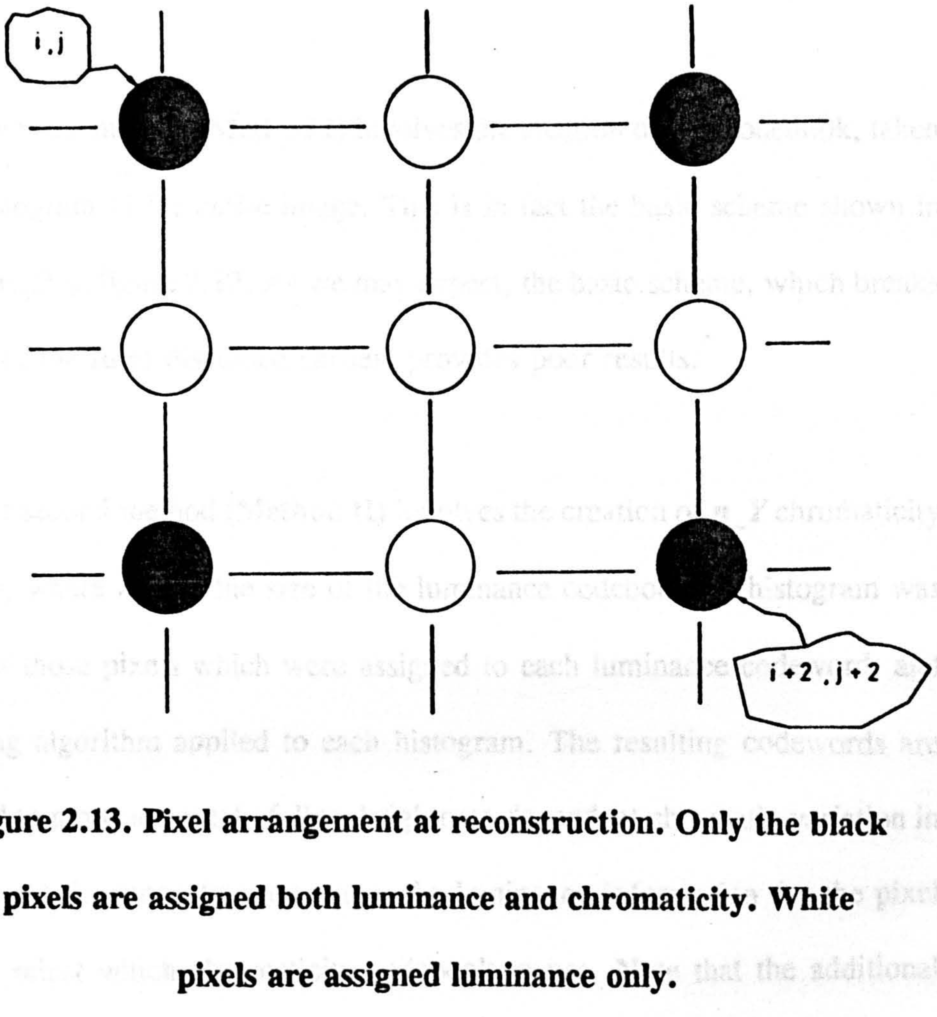


Figure 2.13. Pixel arrangement at reconstruction. Only the black pixels are assigned both luminance and chromaticity. White pixels are assigned luminance only.

2.12. Three Variations on the Quantiser Algorithm

The three variations on the basic outline are now considered. They are in order of increasing complexity and decreasing perceived output distortion. Because the interest in this study is in the colour quantisation, specifically the chromaticity quantisation, the luminance quantisation remained fixed; luminance quantisation involved the creation of a single codebook from the histogram of the entire image (the histogram therefore contains 256 indices).

The first method (**Method I**) involves the creation of one codebook, taken from a histogram of the entire image. This is in fact the basic scheme shown in the flow graph in figure 2.12. As we may expect, the basic scheme, which breaks a number of the rules discussed earlier, provides poor results.

The second method (**Method II**) involves the creation of ***n_Y*** chromaticity codebooks, where ***n_Y*** is the size of the luminance codebook. A histogram was created for those pixels which were assigned to each luminance codeword, and a clustering algorithm applied to each histogram. The resulting codewords are anticipated to more accurately follow brightness-dependent chromatic variation in the image. At the reconstruction stage, the luminance information for the pixel is used to select which chromaticity codebook to use. Note that the additional overhead of multiple chromaticity codebooks is small compared to the size of the output image.

The third method (**Method III**) also involves the creation of ***n_Y*** chromaticity codebooks, where *n_Y* is equal to the number of (*m_Y* x *m_Y*) blocks of pixels in the image. Because all the test images were 256 x 256, *m_Y* is equal to 64 (256/4). This method was devised to exploit position-dependent chromatic variation in the image. The block number at the reconstruction stage is used to select the chromaticity values.

The three algorithms were implemented and tested on three images. These were frames taken from the videoconferencing test sequences mentioned earlier. The sequences are called "Miss America", "Split Screen" and "Trevor White". Figures 2.14, 2.15 and 2.16 portray the test images. It is apparent that these images vary considerably in their content and complexity. "Miss America" contains by far the least complexity in luminance variation, but the face, with its many subtle changes in flesh tones, occupies a large part of the image, and the pink of the model's sweater is fairly saturated. The backing curtain in the "Split Screen" image seems fairly equiluminous but in fact contains a considerable variation in intensity. The majority of colours in this image are unsaturated, however. The "Trevor White" image is by far the most complex; the texture in the curtain and in the shirt is difficult to quantise without distortion. Note also the saturated colours in the red polka-dot handkerchief in the model's pocket.

The experimental conditions under which all images were viewed are described in appendix A. The details of these conditions will be discussed further in this and other chapters.

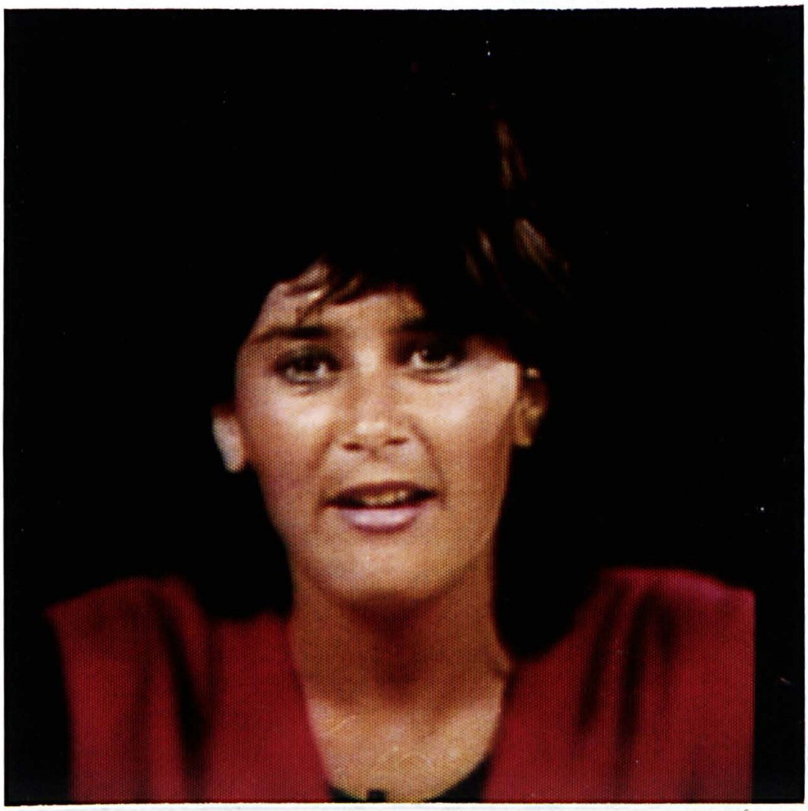


Figure 2.14. Original digitised frame from the video codec test sequence "Miss America". The image resolution is 256x256 pixels.

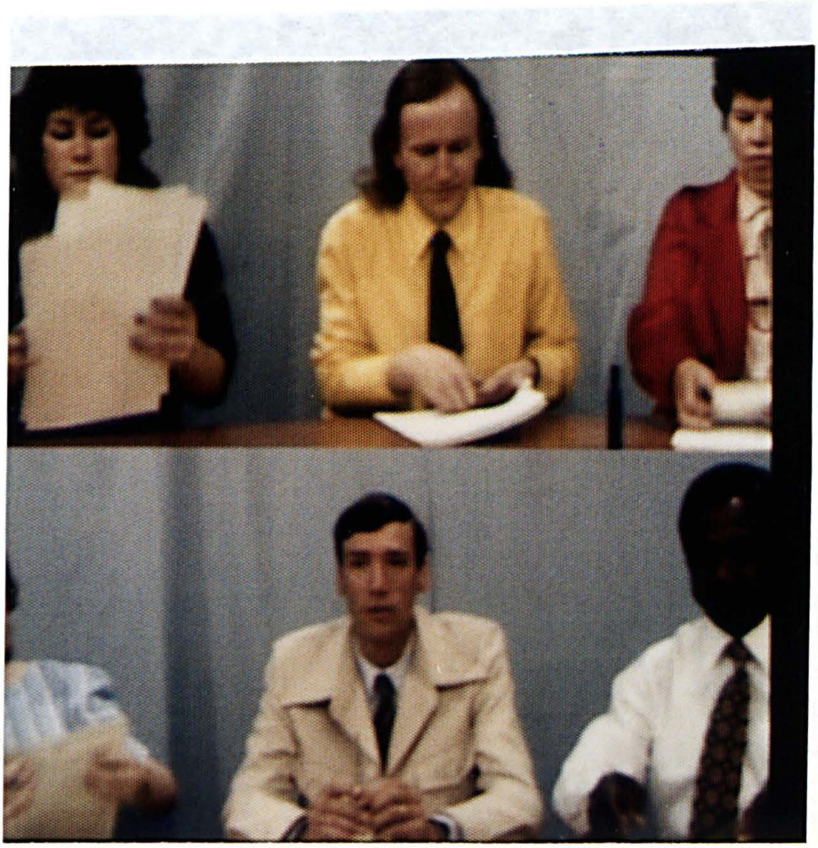


Figure 2.15. Original digitised frame from the video codec test sequence "Split Screen". The image resolution is 256x256 pixels.

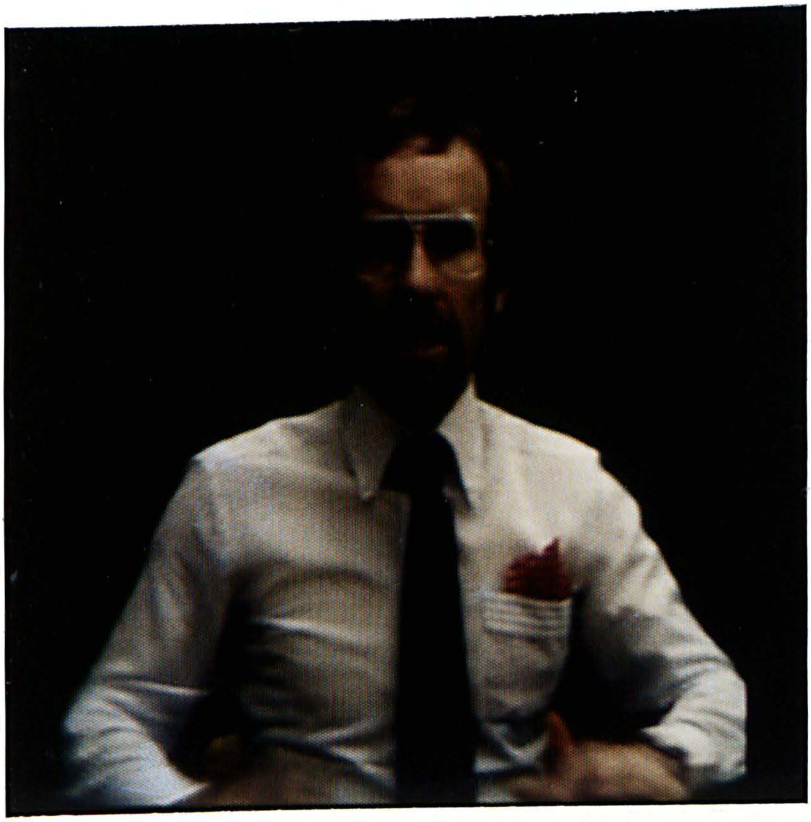


Figure 2.16. Original digitised frame from the video codec test sequence "Trevor White". The image resolution is 256x256 pixels.

Figures 2.17, 2.18 and 2.19 are the results obtained from Method I. It is apparent that there is a considerable distortion introduced by this method. In figure 2.17, the models's face has acquired a slightly bluish tint, and the variety of tones in the face has been replaced with one or two hues. The model's lipstick has vanished. In figure 2.18, the red colour of the jacket worn by the model in the upper right has become subdued, as has the red in the handkerchief worn by the model in figure 2.19. Because the clustering scheme operates in a statistical fashion, these colours, which lie toward the outside of the chromaticity chart, are likely to be rare in the single histogram of the image, hence a symbol has not been found close to the correct chromaticity. In the case of figure 2.17, the bluish appearance of the face is induced by the strong pink of the model's sweater, thus leading the algorithm to find chromaticites with little green content, necessary for successful reproduction of flesh tones.

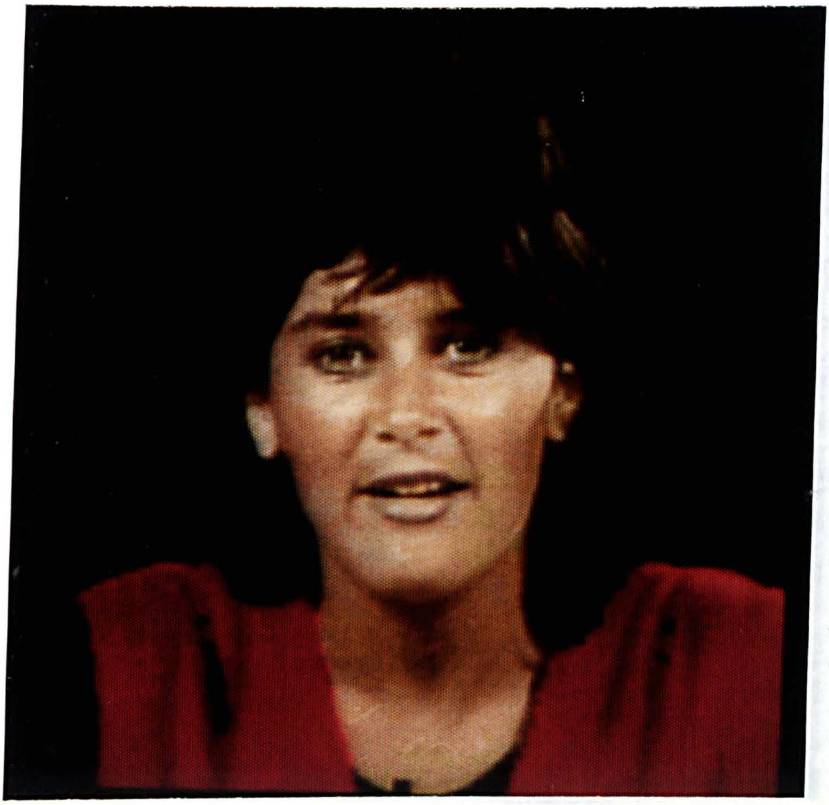


Figure 2.17. Quantisation performed on frame from video codec test sequence "Miss America" using quantiser Method I.

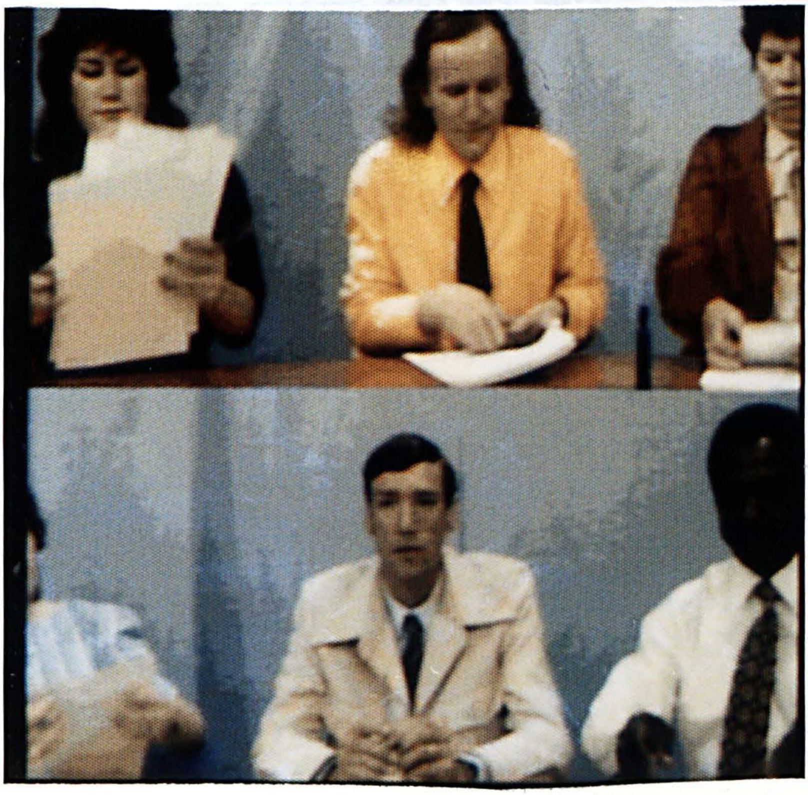


Figure 2.18. Quantisation performed on frame from video codec test sequence "Split Screen" using quantiser Method I.

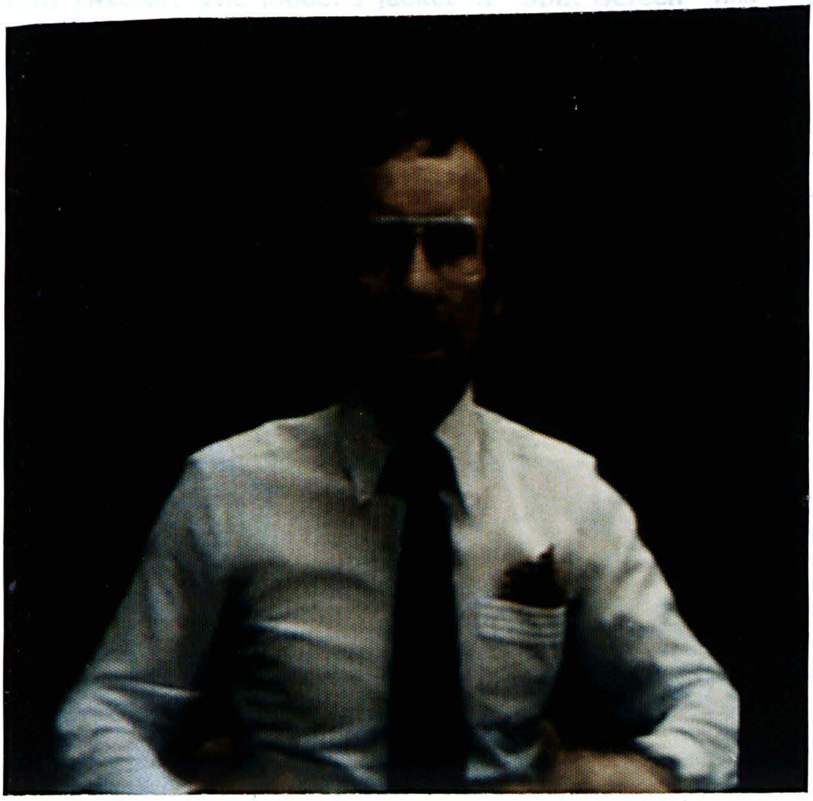


Figure 2.19. Quantisation performed on frame from video codec test sequence "Trevor White" using quantiser Method I.

The results from Method II, shown in figures 2.20, 2.21 and 2.22, are more favourable. The "Miss America" frame contains a more life-like face and a more vibrant sweater. The model's jacket in "Split Screen" has regained its original red colour. Note, however, that "Trevor White" still has a rather muddy brown-coloured handkerchief. Apparently, then, the statistics engine in this scheme is still failing to pick up the rare colours in the handkerchief. Many of the pixels in the model's shirt are obviously assigned the same luminance codeword, hence the resulting chromaticity codewords tend to the achromatic centre of the chart. Figures 2.23 and 2.24 show the variations in the size of the chart at various luminance values (from McColl and Martin [49]). Comparing these contours with those shown in figure 2.25, which show the variations in the size of equiluminous planes through the RGB colour cube, one can see that at the high luminance, the chromaticity chart is restricted. Unless the "sensitivity" or resolution in the chart can be varied with luminance, there will inevitably be a loss of colour accuracy after clustering at high luminance levels.

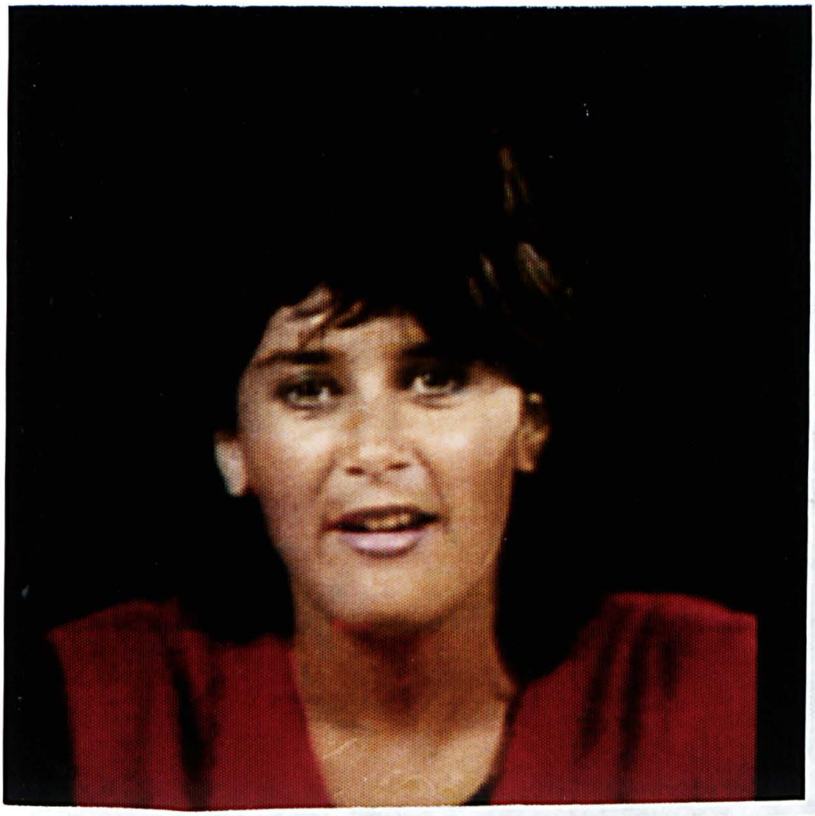


Figure 2.20. Quantisation performed on frame from video codec test sequence "Miss America" using quantiser Method II.

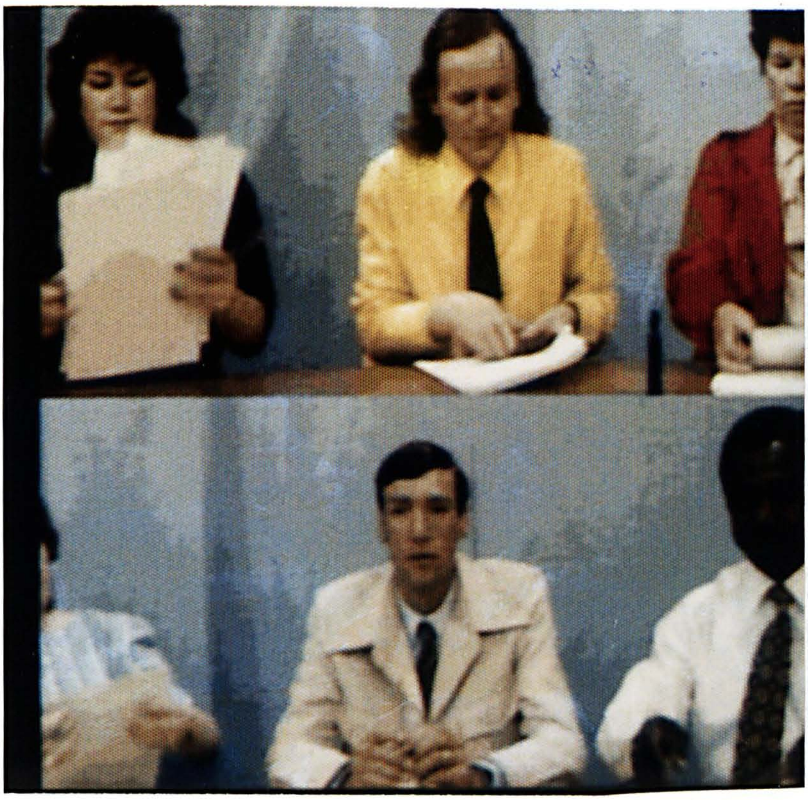


Figure 2.21. Quantisation performed on frame from video codec test sequence "Split Screen" using quantiser variation Method II.

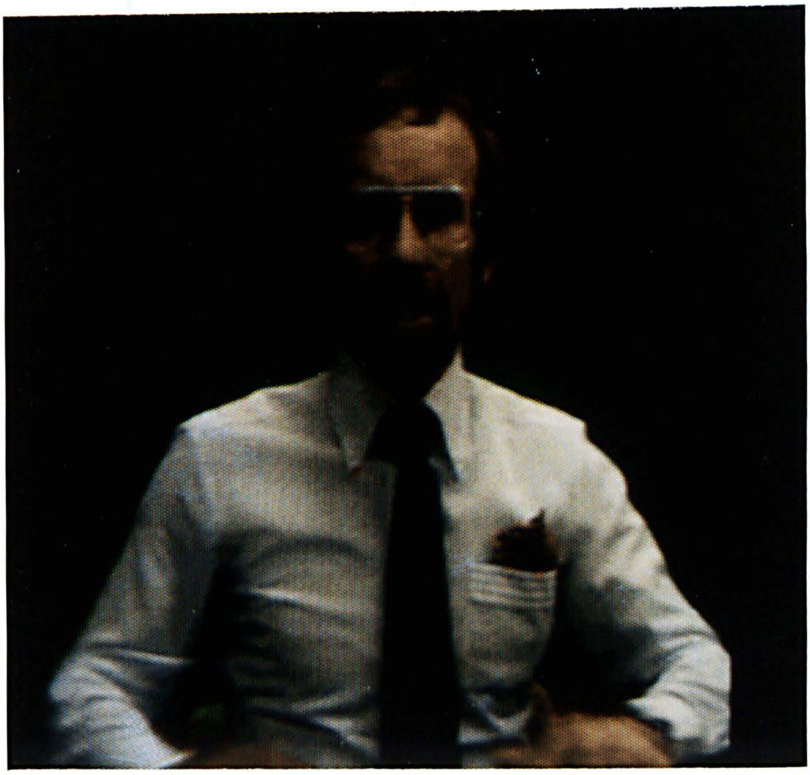


Figure 2.22. Quantisation performed on frame from video codec test sequence "Trevor White" using quantiser variation Method II.

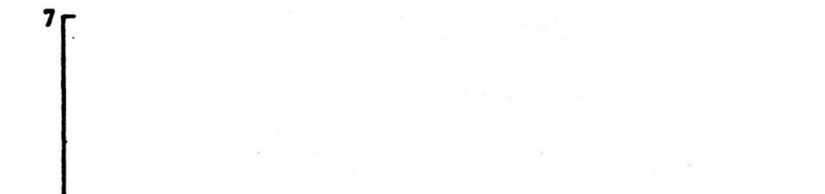


Figure 2.23. Variation of the maximum area on MacAdam's geodesic chromaticity chart as luminance value varies.

(Source: McColl and Martin [49])

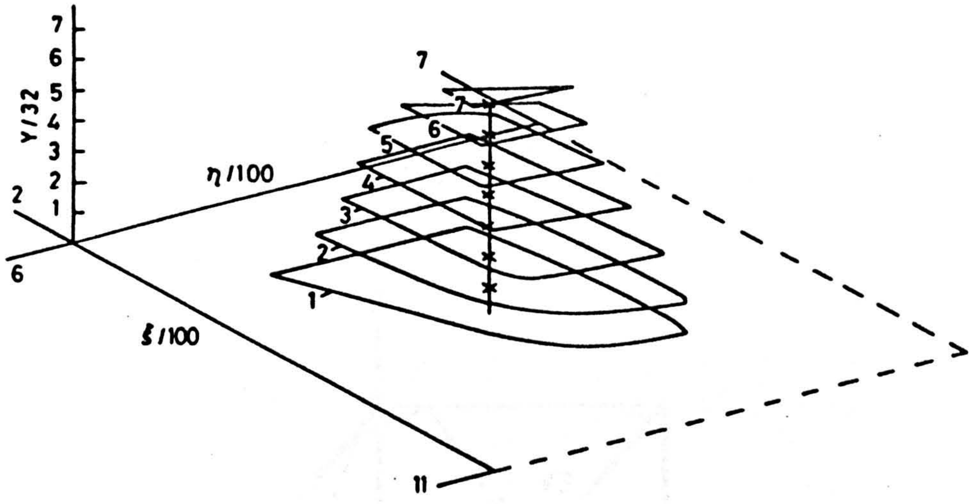


Figure 2.24. 3D plot of available chromaticities on MacAdam's geodesic chromaticity chart as luminance varies.

(Source: McColl and Martin [49])

The results from Method III are shown in figures 2.26, 2.27 and 2.28. It would be fair to say that these are the closest to the originals from an observational standpoint. "Miss America" has regained her flesh tones and her lipstick. The model in "Split Screen" has regained the full colour of her jacket. "Trevor White" has his red handkerchief back. The spatial-dependence of the chromaticity quantisation therefore seems stronger than the luminance-dependence, at least for these test images.

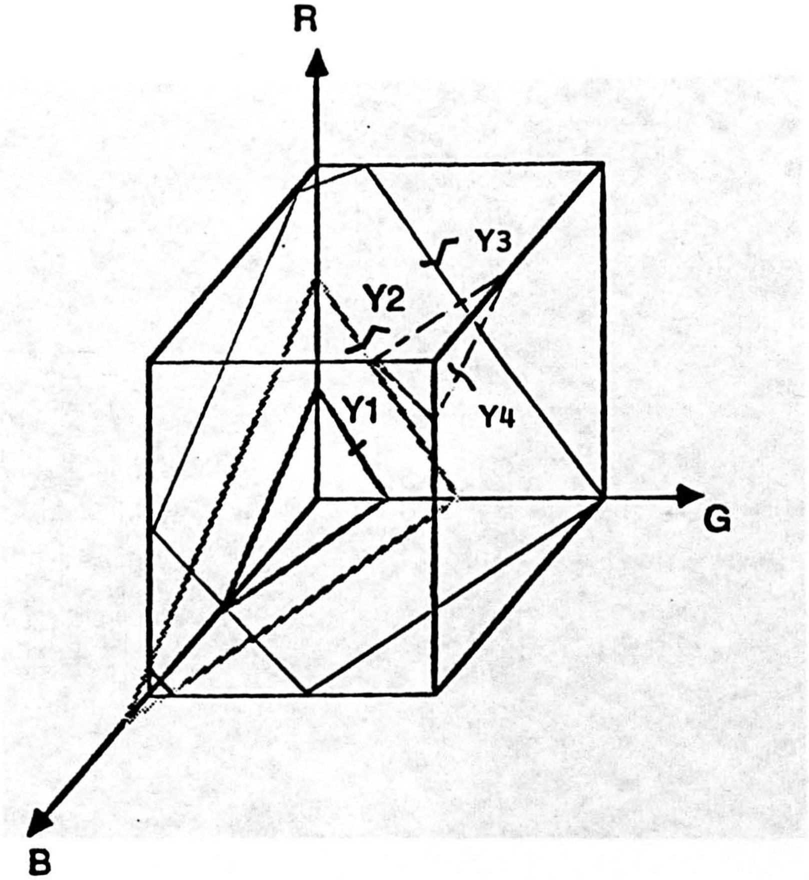


Figure 2.25. Plot of the intersection of luminance planes through the RGB colour cube. Y4 > Y3 > Y2 > Y1.

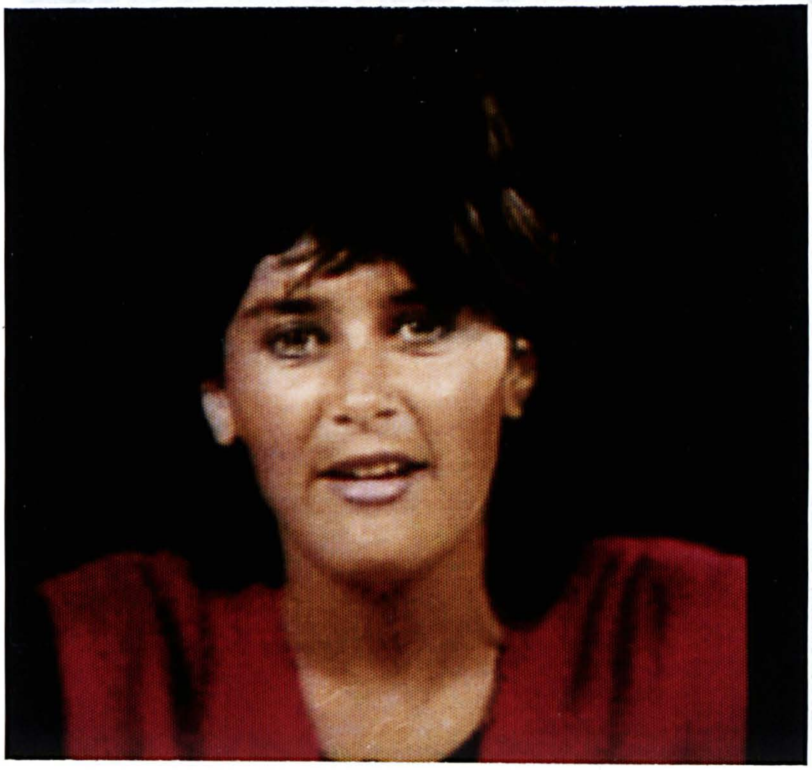


Figure 2.26. Quantisation performed on frame from video codec test sequence "Miss America" using quantiser Method III.

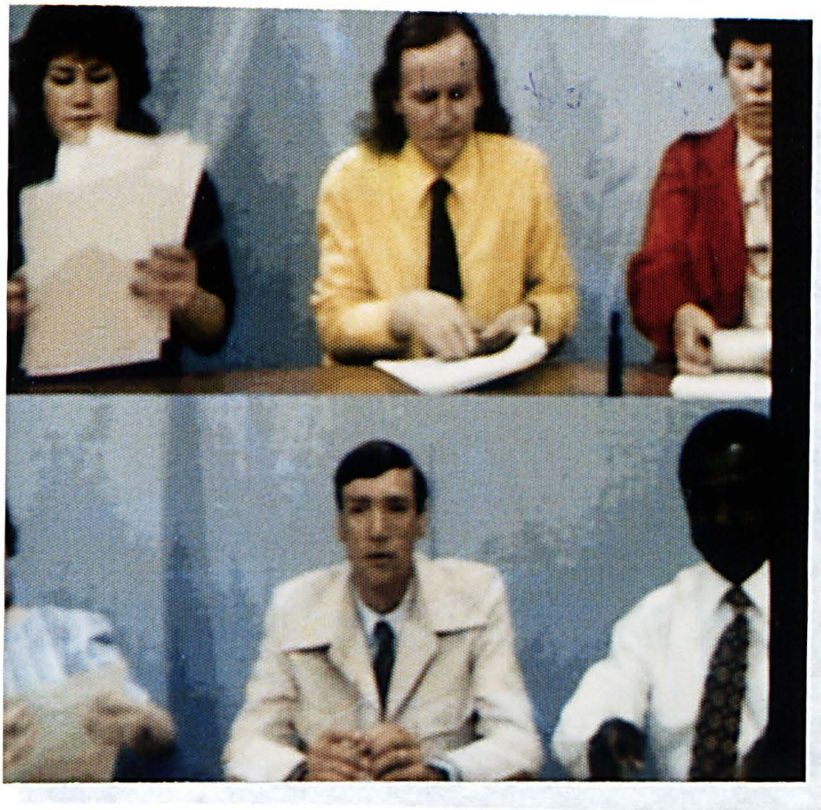


Figure 2.27. Quantisation performed on frame from video codec test sequence "Split Screen" using quantiser Method III.

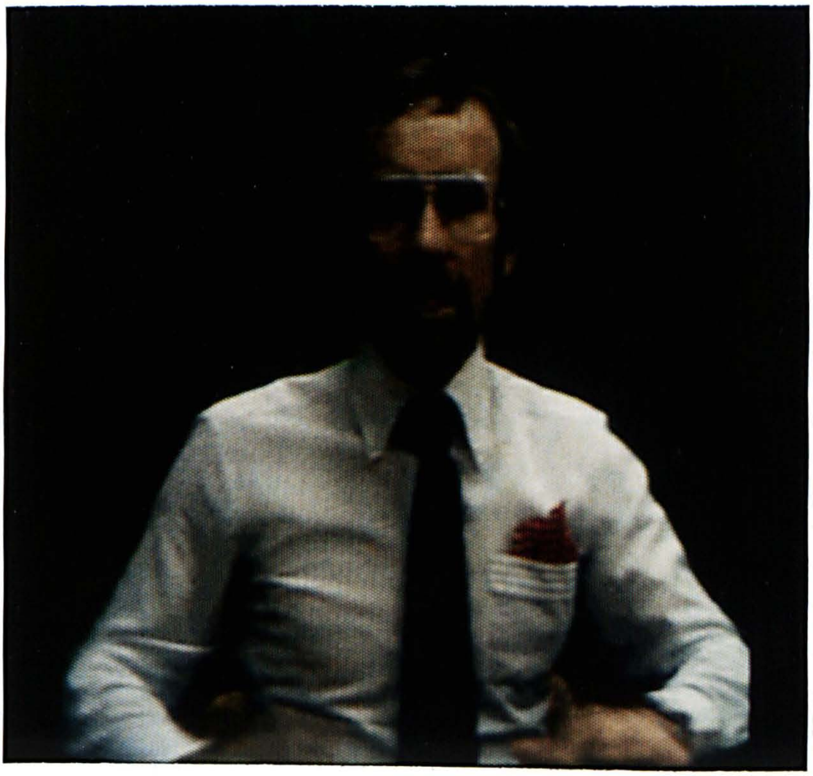


Figure 2.28. Quantisation performed on frame from video codec test sequence "Trevor White" using quantiser Method III.

2.13. Discussion of Results

One important, and "obvious" aspect of the quantisation, which has been neglected thus far, is that for all the methods, the contouring in both the curtain of "Split Screen", and to a lesser extent in the shadows on the face of "Miss America", indicate that the luminance quantisation has not properly succeeded. In fact, 32 levels, or 5 bits, are needed to remove these false contours. The contouring is not apparent in the "Trevor White" image because of the texture in the curtain, which contains a high spatial frequency content. The eye's sensitivity to image content and change decreases with increasing spatial frequency, hence the distortion is not perceived. This will be discussed more extensively in the next chapter.

What about a quantitative distortion measure, and its relation to the perceived distortions introduced by these quantised schemes ? Taking the MSE of the quantised chromaticities from each method, the results are shown in table 2.1.

In order to determine whether the MSE is in fact in good agreement with subjective measurement of colour image distortion, the Mean Absolute Error (MAE),

$$MAE(x,\hat{x}) = \frac{1}{N}\sum_{i=0}^{N-1}|x_i - \hat{x}_i| , \qquad (2.21)$$

was also calculated for each result given above. While it was stated that within the MacAdam geodesic a Euclidean distance should be valid and meaningful, it is unlikely that the visual system can compute Euclidean distances. The results for the MAE statistic are given in table 2.2.

Quantisation	Miss America MSE	Split Screen MSE	Trevor White MSE
Method I	78	73	31
Method II	57	30	26
Method III	49	19	19

Table 2.1. Chromaticity MSE for Quantisation Methods I, II and III on test images.

There is a corresponding agreement with visual assessment of the output images and the distortion measures shown in the tables. Taking the square roots of the numbers given in table 2.1. provides the RMSE, which is on the order of the MAEs given in table 2.2, but, from the previous paragraph, it is unlikely that the visual system is capable of computing square roots. Unfortunately, a multi-subject test, using the "1 to 5" scale for subjective image quality assessment (see Pratt [39], for example, for a discussion of this scale), could not be performed under the laboratory conditions available. Such a test would perhaps have provided a more absolute correlation between numerical distortions and subjective distortions. For these techniques then, the MSE and MAE seem to perform adequately (and similarly) in determining image quality, even though such an error measure does not take into account the magnitude of local distortions.

Quantisation	Miss America MAE	Split Screen MAE	Trevor White MAE
Method I	7	5	4
Method II	6	4	4
Method III	5	3	3

Table 2.2. Chromaticity MAE for Quantisation Methods I, II and III on test images.

It is noteworthy that Limb [112] finds that the RMSE agrees well with subjective distortion, although he finds that local measurements across the image perform slightly better and errors are less visible close to edges. This effect, together with the significant masking of chrominance (and hence chromaticity) errors that can occur at luminance edges (see, for example, Limb [26] or Wysecki and Stiles [35]) implies that chromaticity distortions will not be equally visible, and an improved measure may be found by using the scene luminance signal to weight the measure.

The performance of the variations on the algorithm described in this chapter compares favourably with similar schemes found elsewhere in the literature, such as in Kurz [17], Stevens [24] and Heckbert [33]. These schemes did not necessarily achieve the same compressions, but it is felt that the results presented here improve on the others either in the image quality at the same compression rate, or in compression achieved for similar image quality, or both. In addition, none of the other schemes make an attempt to quantify distortions in reconstructed images.

2.14. Summary

In summary, an image quantisation technique has been designed and implemented with a view to minimising perceived distortion in the output image.

Of the three schemes presented, at least one seems to perform well on the range of test images available, at rates around 4-5 bpp, corresponding to a compression of about 6:1. What could be done to improve these schemes, in order to decrease distortion and increase compression? Improvements (reduction of) the distortion introduced by this quantisation scheme could be achieved in a number of areas. First, the clustering algorithm could be improved. Often, the algorithm finds a set of codewords which are really not the global optima, because of the starting point of the algorithm and the nature of the data. Overcoming this drawback in the k-means iteration is not trivial and may not be tractable [50]. Introducing interdependence in the luminance and chromaticity quantisations, in order to guarantee realisable colours, is also likely to improve the output, and avoid the use of a "nearest neighbour" search when the quantised vector is unrealisable (which may lead to wasted symbols). Perturbing the centroids to provide such agreement is one possible solution. An extra iteration could be added at each stage to provide such a guarantee. This has not been investigated. A fourth quantisation method would logically incorporate both luminance- and position-dependence on codebook construction, but such a scheme would really be infringing on, and would provide poorer results than, the image coding scheme which will be introduced in the next chapter.

The main interest in the quantisation scheme presented here is in reducing, in a reliable manner, the colour cost of the image. As the results, in general, are visually appealing at a cost of 4.75 bits per pel, this technique has been reasonably successful at achieving a coarse colour image quantisation. The chromaticity spaces visited here will be discussed further in the next chapter.

It was noted in the development of the quantisation process that the use of absolute intensity values provided better clusters than using the logarithm of the intensity. This implies a linear relationship between grey-level and perceived brightness (this may be a result of the interaction of monitor gamma and the visual system (see appendix A)). Perhaps what is required, rather than a logarithm computed on every pixel, is the use of a distance measure which has a perceptual analogy and is relates one pixel to another in a more meaningful way than the logarithm (which only relates every pixel to the base *e*). In the next chapter, such a distance measure is introduced.

Chapter 3. Two-Component Colour Image Coding

The introductory chapter contained a brief discussion of the two generations of image processing techniques [27]. The second generation, including algorithms which attempt to mimic and exploit the known features, and in particular the deficiencies, of the visual system, is the subject of this chapter. The context in which this work was conducted was the role of colour in coding textures and images, hence the scope of this part of the research includes those aspects of two-component image coding which apply to visual perception of image distortions and in particular colour image distortions.

This chapter, investigates the implementation of an algorithm to perform the encoding of the data in color images. The algorithm develops from a concept of the structure of an image which is believed to resemble that used implicitly by the human visual system. Emphasis will be placed on the importance of color encoding, the perception of variation in the image content, and on the removal of certain redundancy in the information content of the encoded data. This algorithm is intended to perform some of the functions found in a complete codec suitable for videoconferencing applications at ISDN data rates.

3.1. Introduction

The second generation of image processing algorithms can be considered to be those algorithms which try to exploit known features of the human visual processing apparatus. Prior to 1970, the approaches to image processing for coding and compression involved typically either predictive coding techniques, such as DPCM, or transform-coding techniques, such as DCT. Often, the algorithm would be a hybrid of both, with some sort of additional entropy-compression such a Huffman coding prior to "transmission", whatever that entailed. Generally, by 1970, the compression limit had saturated at about 10:1 [27]. During the 1970s, however, a number of new approaches, which incorporated known characteristics of the human visual system, achieved considerably higher compressions, up to 50:1 and higher. The first algorithm of the second generation, however, was suggested in the 1950s by Schreiber [28]. His algorithm was known as "Synthetic Highs", and consisted simply of complementary low-pass and high-pass filters, which split the input into low- and high-spatial frequency components. The low-spatial frequency signal was coarsely quantised and coded; the high-pass signal was processed, by thresholding, to extract contours of interest, then coded; at the receiver the signals were combined to produce the reconstructed image. By exploiting the lateral inhibition in the visual system, this technique achieved good compression. However, the thresholding operation was poorly arranged, hence the results were either too much lost texture, or too low a compression. This technique did, however, precede the second generation by over ten years. A number of more recent techniques to exploit visual system characteristics are based on this two-component image model.

3.2. Discussion of Image Models

The image model which was decided as the basis for the project is one suggested by, among others, Mason [51], Kocher [52], and Silverman [53] in the early 1980s. This two-component model attempts to consider the image as a scene which may be divided into a series of connected regions of nearly uniform values, known as *texture*, separated by abrupt changes in the signal, known as *edges* or *contours*. These features are the two components of the model. Rather than simply coarsely quantising the signal in the textured areas of the scene, the model generally attempts to describe this data with a polynomial surface function of fixed degree. The contour extraction and transmission is, even today, little changed from Schreiber's in its basic operation: however, the work of those such as Marr and Hildreth [54], and Canny [55], have provided far superior identification of image contours, hence the reconstructed quality and compression have both considerably increased, allowing better preservation of the edge structure, which makes the image appear considerably more "in focus".

There are a number of reasons why the current region-modelling algorithms can be improved. If we consider the content of example regions isolated from the sort of images which we want to compress, and identify those features which are likely to be lost by the use of a parametric region model, then the task of devising a replacement will be simpler. Consider the regions shown in figures 3.1 and 3.2. The first region is taken from a segmented frame from the "Miss America" sequence; the second is taken from the "Trevor White" sequence.

Parametric region models generally incorporate zero-, first- and second-order polynomials as the possible matches for the region content. The zero-order model simply extracts the mean value of the signal in the region. The first-order model extracts a mean value and a fixed gradient across the region, hence there are three parameters. The second-order model extracts the following coefficients to describe the signal $I(x,y)$:

$$I(x,y) = ax^2 + by^2 + cxy + dx + ey + f \qquad (3.1)$$

Hence, the zero- and first-order models simply set some of the above coefficients (a-f) to zero. The most appropriate model is then chosen by finding the minimum error associated with each polynomial surface approximation, such as the MSE, and considering the transmission restrictions on the system. The drawbacks to using this system, apart from descriptive accuracy, are that the coefficients themselves are real valued, and often vary over many orders of magnitude [56], which make the design of descriptive codes difficult. Because of the considerably higher overhead of the second-order model over both the others, in practice only a few regions from each frame can be described using a second-order model so that transmission costs can be kept low enough. Additionally, it is often found that the first-order model provides results which are only slightly more appealing visually than the zero-order model [56], but at three times the cost.

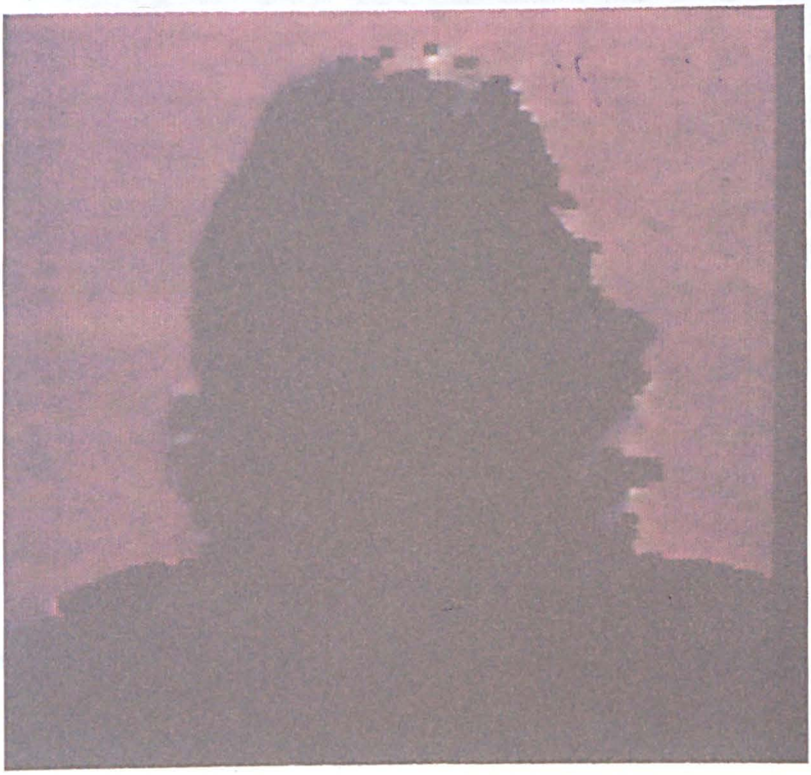


Figure 3.1. Region taken from a segmented frame in the test sequence "Miss America".

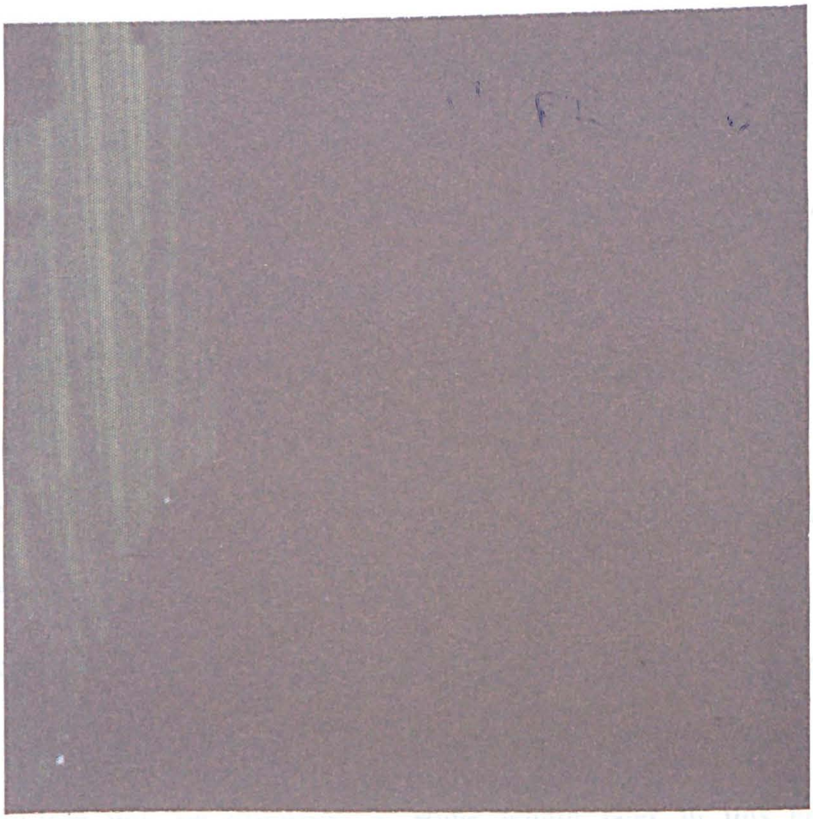


Figure 3.2. Region taken from a segmented frame in the test sequence "Trevor White".

Referring to the figures 3.1 and 3.2, it is apparent that the visible textures in each region will be completely lost by a second-order polynomial description. In order to model these features, the region model will have to take into account local variations as well as global ones. Recently, Julesz, Haralick and others have developed more sophisticated models of image textures, which show the promise of describing texture as a mosaic or repeated tile, with an associated skew or some statistical measure of variation. This type of region model is the subject of chapter 4. For the coder developed in this chapter, the investigation of a suitable region model centred on the use of local-operator models such as pyramidal coders. It was noted by Kunt [27] that because pyramidal coders in general combine predictive and transform coding, that they are in fact hybrid first-generation techniques. However, they possess a structure similar to that of the human visual system, so can be considered as belonging to both generations.

There are two classes of pyramidal coders: the Laplacian pyramid of Burt and Adelson [57], and the quad-tree of Wilson [58], Tanimoto [59] and others. The pyramid or quad-tree forms a structure in which the image is represented at many resolutions, hence the "local" nature of the image varies within each layer. The quad-tree will be discussed at some length later in this chapter. The algorithm developed by Burt and Adelson [57] involves repeatedly extracting image Laplacians, by subtracting Gaussian low-pass filtered versions of the image from itself, saving a coarsely quantised Laplacian image and a half-resolution (in both x- and y-dimensions) low-pass image, then repeating the process until the lowest level of resolution has been reached. In order to reconstruct the image, the saved low-pass image would be interpolated to twice its size, then combined with the saved Laplacian, or "edge" image, then interpolated again, and this process would be repeated until the reconstructed image is the same size as the original. By conceiving the initial image as the bottom level of an image pyramid, then halving the resolution each time is analogous to progressing up the pyramid, with each higher cell containing a low-resolution description of the cells immediately "below". Hence, the algorithm essentially involves the transmission of a series of error images at differing resolutions. There is evidence that the human visual system provides multiple-resolution filtered images from the retinal signal, hence this algorithm elegantly mimics this process.

3.3. The Quad-Tree - Motivation and Use

The motivation for the use of the quad-tree for region-coding is three-fold:

1. The quad-tree has the promise of providing significantly more information than a parametric region model.
2. The quad-tree structure is simple, which permits easier software and hardware design.
3. The data to be coded are regions, which are assumed to possess few sharp discontinuities, i.e. the correlation coefficient between pixels is high. It is expected that few levels of the quad-tree would need to be transmitted in this case.

Additionally, it is important to remember that we are seeking to successfully model and code image regions, not images themselves. The input to the coder will be a previously segmented image, hence there is no need to generate a complex, edge-finding algorithm as part of the coder.

The quad-tree is a hierarchical, or pyramidal, data structure, in which the data resolution halves in both dimensions as we proceed "up" the tree. The "top" of the tree contains only one data value, or *node*. It is connected to four nodes at the next level going "down" the tree; these nodes are referred to as its *children*; it is their *parent*. The children are in turn each connected to four children one more level down, and so on until the lowest level in the tree, in which the nodes have no children. These nodes are referred to as the *leaves* of the quad-tree. Hence, if the tree is *complete*, i.e. all nodes except the leaves have four children, and there are N^2 leaves, then there will be $log_2(N)+1$ levels in the tree, so the levels can be numbered, from the base to the root, as $\mathbf{0,1,2,..}\boldsymbol{m}\mathbf{,..,}\boldsymbol{n}$, where $\boldsymbol{n}$ equals log_2N, and at level $\boldsymbol{m}$, the quad-tree will have M^2 nodes, where M equals 2^{n-m}.

In order to generate a series of images at differing resolutions in the quad-tree, the method is to assign the leaves of the tree the values of the pixels in the input image. We assume for now that the image is square, of side N. By assigning the parent of each group of four (2x2) leaves their average, and repeating at the next level, we arrive at the top of the tree where the root contains the average value of the whole image. It is interesting to note at this point, that if we are attempting to estimate the value of a node at level **m**, the best estimate (in a MSE sense) will always be the value of the parent node [58] (from the orthogonality principle [60]). Thus the best estimate is simply the average of the children.

3.4. The Use of Incomplete Quad-Trees

It is apparent that the structure of the quad-tree is likely to be very different from that of an image region, which may be highly convoluted, such as the example shown in figure 3.3. In order to model and code the region, then, the quad-tree must be somehow fitted over the region. The only practical solution is to find the maximum orthogonal dimension of the region, which is supplied from the output of the segmentation algorithm, and centre the appropriately sized quad-tree over the region, as shown in figure 3.4 (from McColl and Martin [61]). Referring to this figure, it is likely that the quad-tree will be incomplete. The averaging process, and hence the estimation algorithm, must be modified to take this into account. Hence, for any node in the quad-tree above the leaf level, i.e. at any level $l > 0$, the value can be expressed as

$$\bar{s}_{i,j,l} = \frac{\sum_{ii=0}^{1}\sum_{jj=0}^{1} \bar{s}_{2i+ii,2j+jj,l+1} \cdot t_{2i+ii,2j+jj,l+1}}{\sum_{ii=0}^{1}\sum_{jj=0}^{1} t_{2i+ii,2j+jj,l+1}}, \quad l = 1,2,..,\log_2 N \qquad (3.2)$$

where the bars indicate averages, and the denominator indicates the number of leaf nodes accessible from this node. $t_{0,0,0}$ equals one or zero. This number then will always be a multiple of four if the tree is complete, but could be zero otherwise; in this situation, the upward averaging process must check before a division is attempted. This modified quad-tree structure must store at every node, as well as the average, the number of leaves below it.

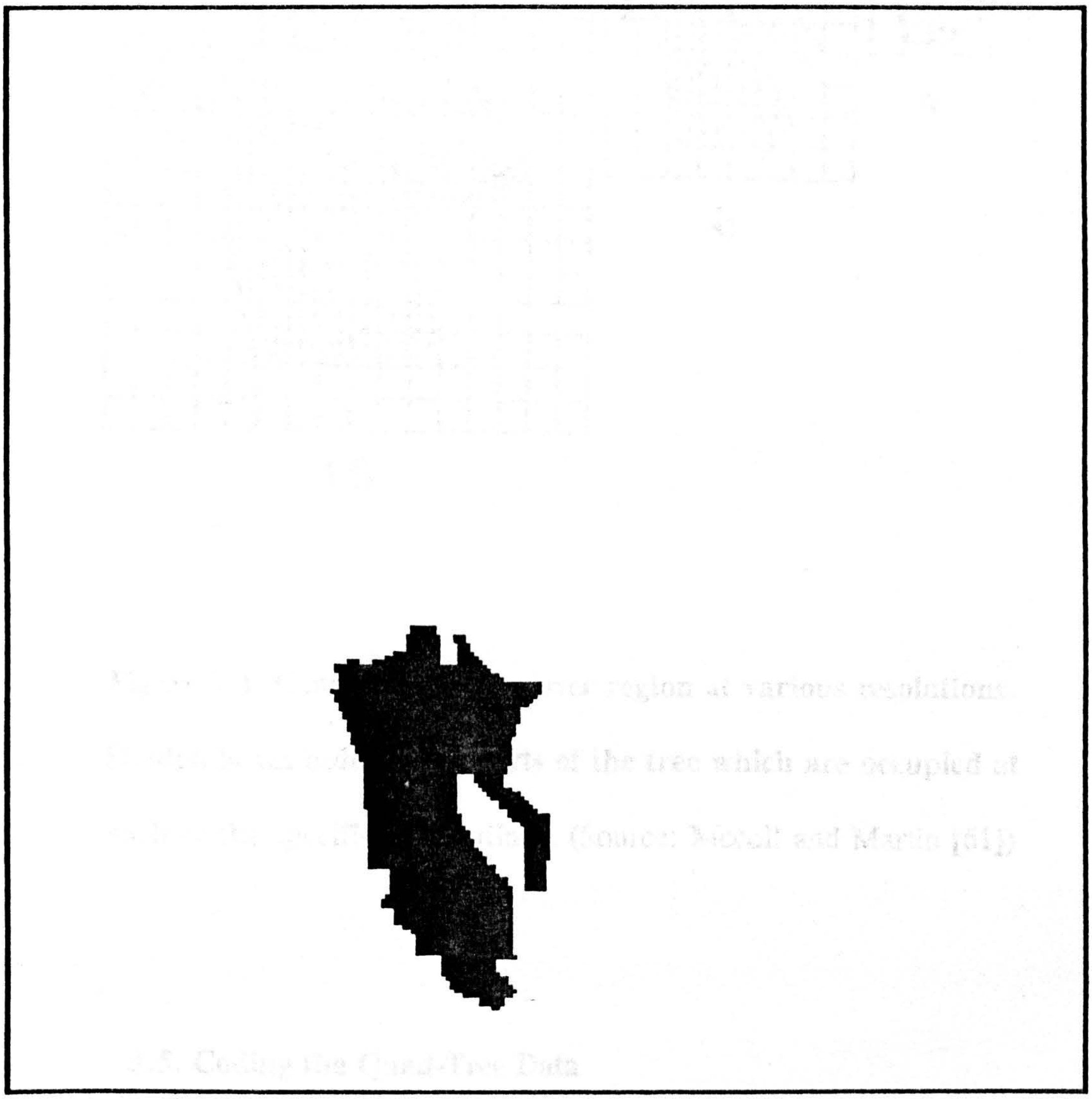


Figure 3.3. Dimensions of highly convoluted region taken from segmented frame in test sequence "Trevor White".

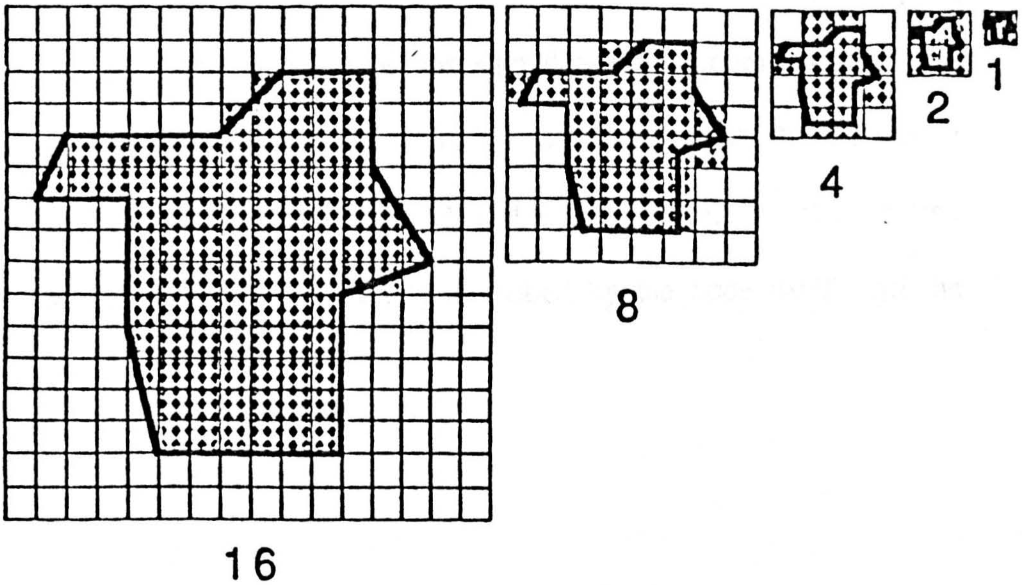


Figure 3.4. Centred quad-tree over region at various resolutions. Shaded boxes indicate the parts of the tree which are occupied at each of the specified resolutions. (Source: Mccoll and Martin [61])

3.5. Coding the Quad-Tree Data

Once the quad-tree has been constructed by upward averaging, the region coding process can take place. This is normally a recursive process. Each node, from the root through its children and so on, is visited in turn. If a criterion, which is usually some error or distortion measure, is exceeded for that node, then each of its children will be visited and the process will continue, otherwise, this downward process will terminate at that node. The further the process continues through the tree, the more data is likely to be generated. For any region, then, the amount of generated data should correspond with the complexity of, or activity in, the region. The simplest criterion is one in which the difference between the node and its children is found, and if this is zero, then the sub-tree with this node as its root is completely described by the node itself - all the descendants have equal values.

3.6. Coding the Chromatic Content of the Region

Since the topic of this research is perceptual distortions and in particular, colour distortions, it is important to compare the approach to coding colour region data found in the literature. Of the papers found, the RGB colour space was used for the coding, and the parametric model used was identical to that for the luminance. However, it is found in practice that a zero-order model is sufficient to provide acceptable results. Experiments conducted on images from the digitised test sequences bore out this statement. By contrast, the use of a zero-order model on the luminance in every region produces extremely unappealing results. In fact, the visual system is far less sensitive to colour changes than to luminance changes [62]. This is demonstrated by examples such as watercolour and ink paintings and children's crayon drawings, in which the sharp outlines of the scene are first drawn, then the enclosed regions are "filled" with the desired colour, using a colour "wash". This phenomenon is often explained as a demonstration that the visual system removes the chromatic component from the visual stimulus early in the pathway to the processing centres, and uses the achromatic signal alone for the scene segmentation. A corollary to this argument is that in most images, a change in pixel colour is almost always accompanied by a change in pixel luminance. although there has been research which indicates that colour changes without luminance changes may be important in some recognition tasks.

Because the experience in this research indicated that an analogy to a zero-order parametric model works well in reproducing the colour in image regions, this simple model was pursued. From the previous chapter, it is apparent that using the RGB triplets to provide an average colour value for the region will not provide the perceived average. The average chromaticity was found instead. In order to provide a reasonable approximation to the perceived average, the chromaticity chart used was either the CIE-UCS or MacAdam chart. Because of the inability of the MacAdam chart to preserve centre-of-gravity, the work centred on the use of a (u_{ave}, v_{ave}) chromaticity pair for each region. The coding of these data will be discussed further in this chapter.

3.7. Threshold Coding of Quad-Tree Data - Image Contrast

The data to be coded for the pixels in each region, then, were the luminance and chromaticity. Because the segmentation data is to be used to provide the edges in the reconstructed image, it was important that the region coding produce a succinct region description which nonetheless provided considerably more region detail, and thence produced more appealing reconstructions, than the parametric region model at similar compression rates. The use of the quad-tree as the structure for the model meant that the coding process would be a function of the properties of the nodes and their descendants. Because it was found that the average chromaticity value for the region is generally sufficient to provide adequate colour in most reconstructed scenes, the contents of the quad-tree would be the luminance only, so the coding process should consider the perceived distortions introduced in incomplete transmission of the quad-tree. As mentioned earlier, the simplest criterion is whether a node exactly represents all those below it. In practice, this coding procedure is insufficient for high compressions, even for region data. Wilson [58] suggested that the compression available from the quad-tree alone is not sufficient for classical image coding. He proposed the use of a DCT on the image together with hierarchical coding, so that the quad-tree encoding operated on spatial-frequency data, with a quantised parent-child difference which terminated when this fell below a set threshold. Clippingdale and Wilson [63], using the quad-tree to provide image enhancement, used a recursive function of the node energy, or variance, at each level in the tree to provide the best estimate of the child and hence drive the parent-child difference function. The energy distribution will not be stationary through the tree, hence using the node energy alone is insufficient to determine activity in the sub-tree. These schemes, however, are used on complete images and it must be remembered that regions are being coded here.

Recall that the visual system has a linear response to image contrast. The Weber fraction definition [40] of image contrast C for a target of uniform luminance against a uniform background is

$$C = \frac{\Delta L}{L} \tag{3.3}$$

where ΔL is the increment or decrement of the target luminance from the uniform background luminance L. Some people have noted that this value is a constant, with reported values ranging from 0.01 to 0.15, depending on the experimental conditions. There is another formula proposed by Michelson [42], used to determine contrast C of periodic patterns such as sinusoidal gratings. It is defined as

$$C = \frac{L_{\max}-L_{\min}}{L_{\max}+L_{\min}}\,. \tag{3.4}$$

where L_{max} and L_{min} are the highest and lowest luminance encountered in the sinusoid. Note that the contrast sensitivity of the human visual system is not a constant but is in fact a function of spatial frequency. The response is somewhat like a low-pass filter, with a small peak prior to the cutoff. The curve, as found by Kelly [64], is shown in figure 3.5 (from Robson [65]). In this diagram, the relationship between contrast sensitivity and spatial frequency is shown for varying temporal frequencies (also measured in cycles/second). This means that, except for very low temporal frequencies, the observer is less sensitive to high-frequency "noise" or image distortion than to errors which have a low spatial frequency. Returning to the choice of distortion criteria for visual fidelity, it is obvious that high frequency noise alone is liable to be less visible than low frequency noise, even if the MSE of the reconstructed image is the same in each case.

The importance of minimising the perceived distortion in the image, or image region, leads us to find, for each node in the quad-tree, some threshold which correlates to a fixed perceived distortion. By using the Weber fraction, it seems that the potential visibility of the sub-tree can be determined for any node: if all the nodes below any node are perceived to be no different from it, then the coding process can terminate. In practice, what is required then is to provide the ratios

$$\frac{\left|\max_{(l,m,n)\in ST_{i,j,k}}(s_{l,m,n}) - \bar{s}_{i,j,k}\right|}{\bar{s}_{i,j,k}} \qquad (3.5a)$$

and

$$\frac{\left| \min_{(l,m,n) \in ST_{i,j,k}} (s_{l,m,n}) \; - \; \bar{s}_{i,j,k} \right|}{\bar{s}_{i,j,k}} \tag{3.5b}$$

where $ST_{i,j,k}$ is the sub-tree whose root is $\bar{s}_{i,j,k}$, ***max()*** is the largest value found, ***min()*** is the smallest value found and | | indicates that the magnitude is taken. If neither of these ratios is found to exceed the level at which this contrast will become just noticeable, i.e.

$$|\max(s) - \bar{s}| < c \cdot \bar{s}; \; |\min(s) - \bar{s}| < c \cdot \bar{s} \,, \tag{3.6}$$

where ***c*** is the "just noticeable contrast" defined by Weber, then the transmission of region data can stop at this level in the quad-tree, otherwise the children must be similarly tested and so on. In order to avoid the recursion inherent in finding the maximum and minimum value in the sub-tree, these can easily be calculated as part of the upward averaging process.

By using the Weber fraction as the threshold criterion, a slightly different approach is being taken to perceived brightness (and perceived brightness difference) than that of the previous chapter. Indeed, there is some argument over this issue, as has been discussed earlier in this paper. There will be further discussion about this issue at the conclusion of this chapter and in chapter five.

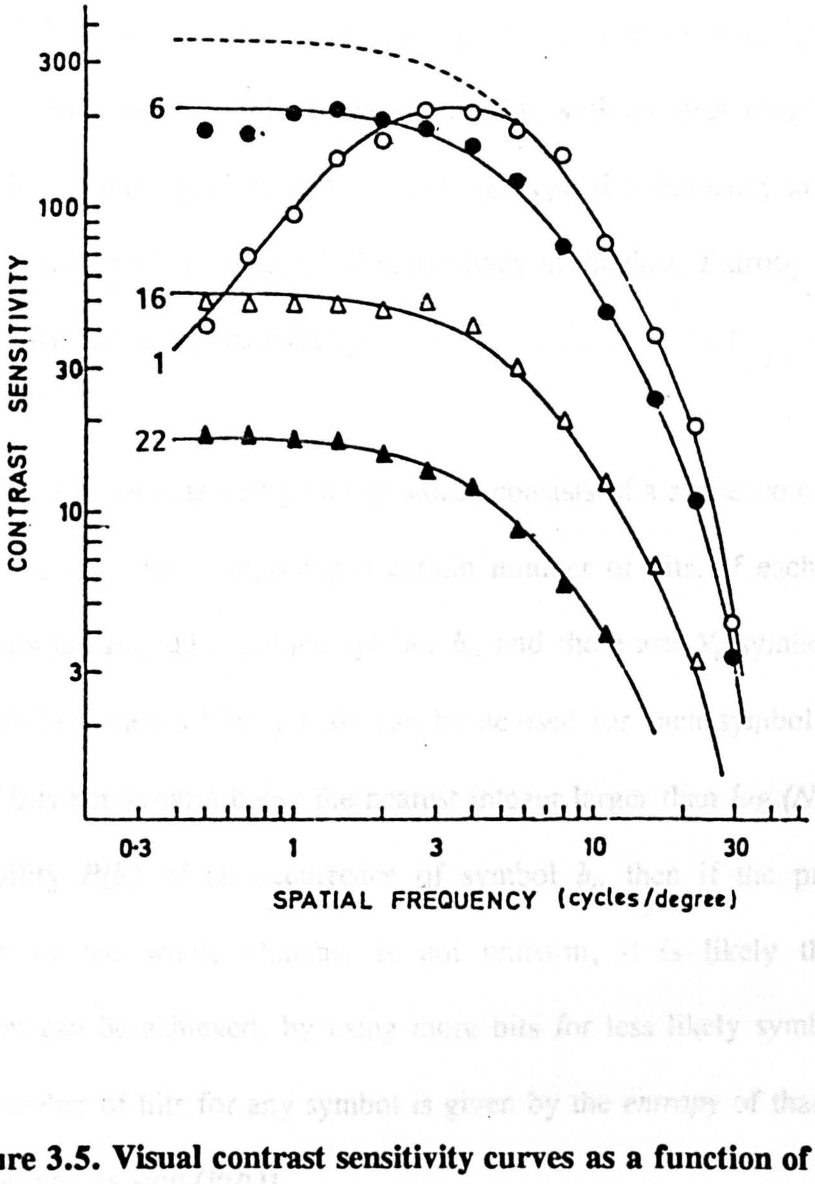


Figure 3.5. Visual contrast sensitivity curves as a function of spatial frequency, for temporal frequencies of 1, 6, 16 and 22 cycles/degree.

(Source: Robson [65])

3.8. Entropy Coding

It was mentioned previously that the best estimate of any node in a quadtree is the parent node. This estimate is the best in the sense of generating a minimum MSE prediction error. When coding the region, it is obviously desirable to send or store as few bits as possible. As well as providing the best reconstruction of the region by removing the perceptual redundancy in the data, we also need to remove any statistical redundancy in the data. Entropy coding is used to remove statistical redundancy.

The stream of data being sent or stored consists of a sequence of discrete data values, each value comprising a certain number of bits. If each possible group of bits is assigned a unique symbol $\boldsymbol{b_i}$, and there are $\boldsymbol{N_b}$ symbols in the symbol alphabet, then a binary code can be devised for each symbol with the number of bits per symbol being the nearest integer larger than $\boldsymbol{log_2(N_b)}$. Given the probability $\boldsymbol{P(b_i)}$ of an occurrence of symbol $\boldsymbol{b_i}$, then if the probability distribution of the whole alphabet is not uniform, it is likely that some compression can be achieved, by using more bits for less likely symbols. The optimum number of bits for any symbol is given by the *entropy* of that symbol, which is defined as $\boldsymbol{-log_2[P(b_i)]}$.

Entropy coding the data stream produces a codeword or codewords by combining the appropriate codes for each of the received symbols. If a technique such as Huffman coding is used [66], this codeword is simply a concatenation of individual codewords selected from a previously calculated alphabet. The alphabet must be computed before the data can be coded, which in general means that the entropy coding will be a two-pass algorithm. The first pass generates the statistics which are used to calculate the codes; the second to construct the codeword. The alphabet must also be transmitted, introducing overhead into the entropy compression. As well as this, Huffman coding also suffers from the fact that an integer number of bits is required for each codeword; in general, then, the Huffman code for each symbol will be greater than the entropy of that symbol. In theory, if the entire sequence of input symbols were taken as a whole, then by finding an alphabet to code all possible sequences of symbols, a Huffman code can be devised which would converge on the entropy as the concatenated codeword becomes large. However, this is clearly impractical as the number of possible multi-symbol sequences would be the symbol size *raised to the power of* the codeword size. One method of accurately approximating the entropy of a sequence is to use *arithmetic* coding [70]. This method is particularly effective if the symbols in the sequence are not independent.

3.9. Arithmetic Coding

In arithmetic coding, instead of forming the output codeword from the concatenation of the codewords associated with the input symbols, the output is a single fraction between 0 and 1 which, in the limit, equals the combined probabilities of the complete sequence. The work by Rissanen and Langdon [67], and Mitchell and Pennebaker [68,69] at IBM during the 1980s has shown that a number of practical problems associated with arithmetic coding are tractable. This has made arithmetic coding more attractive as a tool for the removal of statistical redundancy. The problems are the *infinite precision* problem, which requires that the precision (in bits) of the output codeword is unknown until all the data has been read; the *carry over* problem, which involves the possibility of an arithmetic carry operation propagating through an arbitrarily long sequence of 1's in the codeword; and the *increasing precision* problem, which requires that the precision required from the multiplication of two finite precision numbers is always higher. All these problems prevent the decoder from being able to start decoding the transmitted codeword until the entire sequence is received. The buffering and communication challenges presented were considered insoluble until the IBM results appeared in the literature.

Arithmetic coding can be viewed as a process of repeatedly sub-dividing the real number line **[0-1]**. Initially the line is divided into intervals, the width **(A)** of each interval representing the probability of each symbol, and the starting point of the interval being the sum of all preceding symbol probabilities. This starting point is used as the binary codeword **(C)** for the current state of the sequence. When coding a symbol, any binary value in the corresponding interval **A**, from the starting point for that symbol to the starting point of the next symbol (but not including it), uniquely identifies the given symbol. When another symbol is to be coded, the line **[0-1]** is replaced by that part of the line occupied by the first symbol, a fraction, and subdivided by the alphabet once again, in the same proportion as before. Again, any point within the new interval, from the start of the new symbol to the start of the next, will uniquely identify the two-symbol sequence. An example of this is given in figure 3.6, in this case for an alphabet with symbols **a**, **b** and **c**, whose probabilities are **0.5**, **0.25** and **0.25** respectively. Note that these probabilities correspond to the binary probabilities **0.1**, **0.01** and **0.01**, which accumulate to **0.1**, **0.11** and **1.0**. In this example, the codeword **C** for the symbol pair **bb** will be **0.101** and the interval size will be **0.0001**.

The process of splitting the number line continues until the entire codeword is generated. The length of the interval associated with this completed process will be the product of the probabilities, i.e. $P(i_1,i_2,...,i_n)$, and hence the logarithm of this interval will be the entropy of the sequence. From this, a point can be found on the final interval, with $\lceil -log_2(A) \rceil$ bits, where $\lceil\ \rceil$ is the ceiling operator, which means "nearest integer greater than" and A is the interval size. The decoding process is simply the reverse, with the decoder using the boundaries it knows about to calculate the coded symbols.

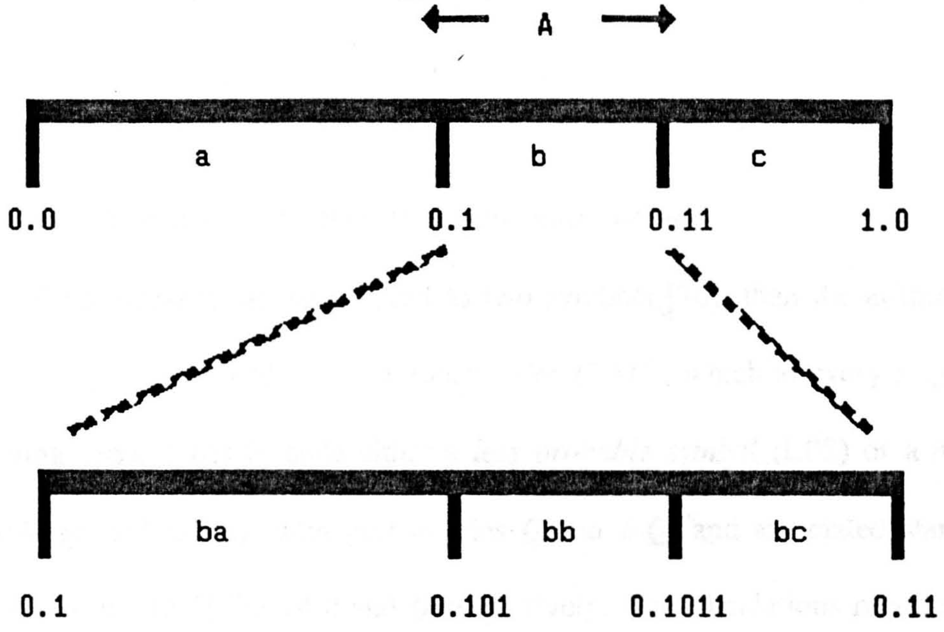


Figure 3.6. Example of arithmetic coding for three-symbol alphabet. Probabilities are P(a)=0.5, P(b)=0.25, P(c)=0.25.

The obvious advantage of arithmetic coding over Huffman coding then is that the probability for the given sequence only is being calculated, rather than the probabilities for all possible sequences. Another important feature of the arithmetic coder is that, if the coder and decoder have attached statistics units or other knowledge of dynamic behaviour, then the various symbol probabilities can change, to reflect *priori* dependence, without affecting the coding or decoding algorithm. With Huffman coding, there would obviously be a need for many alphabets for this to be implemented. (Dynamic Huffman coding [71] does go some way toward solving this problem, but the algorithm is slow and the other limitations of Huffman coding still apply.)

3.10. The Binary Arithmetic Coder and Q-Coder

If the alphabet can be reduced to two symbols [70], then the arithmetic coder is simplified to a *binary arithmetic* coder (BAC), which at every stage in the coding process has to code either a *less probable symbol* (LPS) or a *more probable symbol* (MPS), with probabilities ***Q*** and ***1-Q***, and associated starting positions on the **[0-1]** line of **0** and ***Q*** respectively. The calculations required to create the codeword are simplified, and by modifying ***Q*** at every stage to take account of the changing signal statistics, the coder can be made to follow the local entropy of the symbol sequence. Thus input symbols are rendered as binary events by constructing a binary tree for the alphabet, with number of leaves equal to the number of symbols. Moving through the tree to reach the leaf associated with a particular symbol involves repeated selection of a MPS or LPS, which is the input to the BAC.

The problems which the IBM team solved involved the infinite precision required of the interval *A* on the number line and the increasing precision required when *A* is multiplied. By truncating the result of multiplication, the compression is reduced, but the increasing precision problem is solved; by "windowing" the precision of the current interval to a fixed number of bits, and by "stuffing" additional bits from time to time if a series of 1s are encountered (to prevent a carry greater than the length of the window), the infinite precision problem is solved.

Additionally, Mitchell and Pennebaker designed a DPCM-coding algorithm which they called the Q-coder [68,69], in which the coder and decoder, working in synchrony, need not perform multiplications of cumulative probabilities and intervals. By detecting when the "local" interval gets too small, each unit left shifts the interval A, thereby multiplying by two. These simplifications reduce the operations required to create the codeword from a maximum of two high-precision multiplies and an addition, for an event in the general arithmetic coder, to at most two additions, for a LPS/MPS bit in the (BAC) Q-coder.

The importance of the arithmetic coder's ability to easily code changing statistics is its attraction as far as this study is concerned. For the Q-coder, a fixed set of possible probabilities are used as a lookup table to drive the estimate $\boldsymbol{Q_e}$, a measure of the current LPS/MPS ratio. This value will change to keep track of the local ratio of LPS to MPS events. Because the Q-coder renormalises every time the interval A drops below a fixed value, then if the chosen event which causes this renormalisation is an LPS, the coder assumes that the estimate $\boldsymbol{Q_e}$ is too low, so the next higher value of $\boldsymbol{Q_e}$ is selected and the coder begins to receive LPS decisions. If this continues, then at some point another renormalisation will occur, and the previous Q_e will be selected. In this way, the coder tracks the local entropy of the signal. By choosing many different Q_e values for different nodes in the binary tree for the symbol alphabet, which are known as *contexts*, the coder can dynamically track the entropy associated with each node, and hence the entropy of the signal as a whole. The design of the table of Q_e values can be made to reflect the granularity of the signal statistics and its general shape, e.g. Laplacian or Gaussian.

3.11. Region Coding - Quantisation

Once the quad-tree is constructed, the tree traversal, with each node tested to see if it meets the perceived distortion criterion, will result in production of quantised parent-child differences. The root node is coded separately. The distributions of the quantised parent-child differences for all of the regions of a selected frame from each of the three test sequences are shown in figures 3.7, 3.8 and 3.9, on a y-axis log-scale, for a contrast threshold value of 0.01. The horizontal axis is in integer multiples of c (All indices with entries of 1 are also shown.) As we expect, the distribution is Laplacian in each case. Because of the ability of the Q-coder to adapt to local entropy changes, the quantiser can be linear. The quantiser form *q()* used here is

$$q(\delta Y,Y,c) = int(\frac{\delta Y}{c \cdot Y}+0.5), \tag{3.7}$$

where ***Y*** is the value of the parent, ***δY*** is the magnitude of difference between the parent and the child, ***c*** is the Weber fraction, and ***int()*** generates the nearest integer less-than-or-equal-to the argument. This quantised value is then used to provide an error of the value of the child node if traversal continues. The "quantised" child estimate is

$$(\delta Y+Y)_q = (q(\delta Y,Y,c) \cdot c + 1) \cdot Y \,. \tag{3.8}$$

In these equations, the ***sign*** of ***δY*** is saved and used to select which codeword to select. It must be included when calculating (3.8).

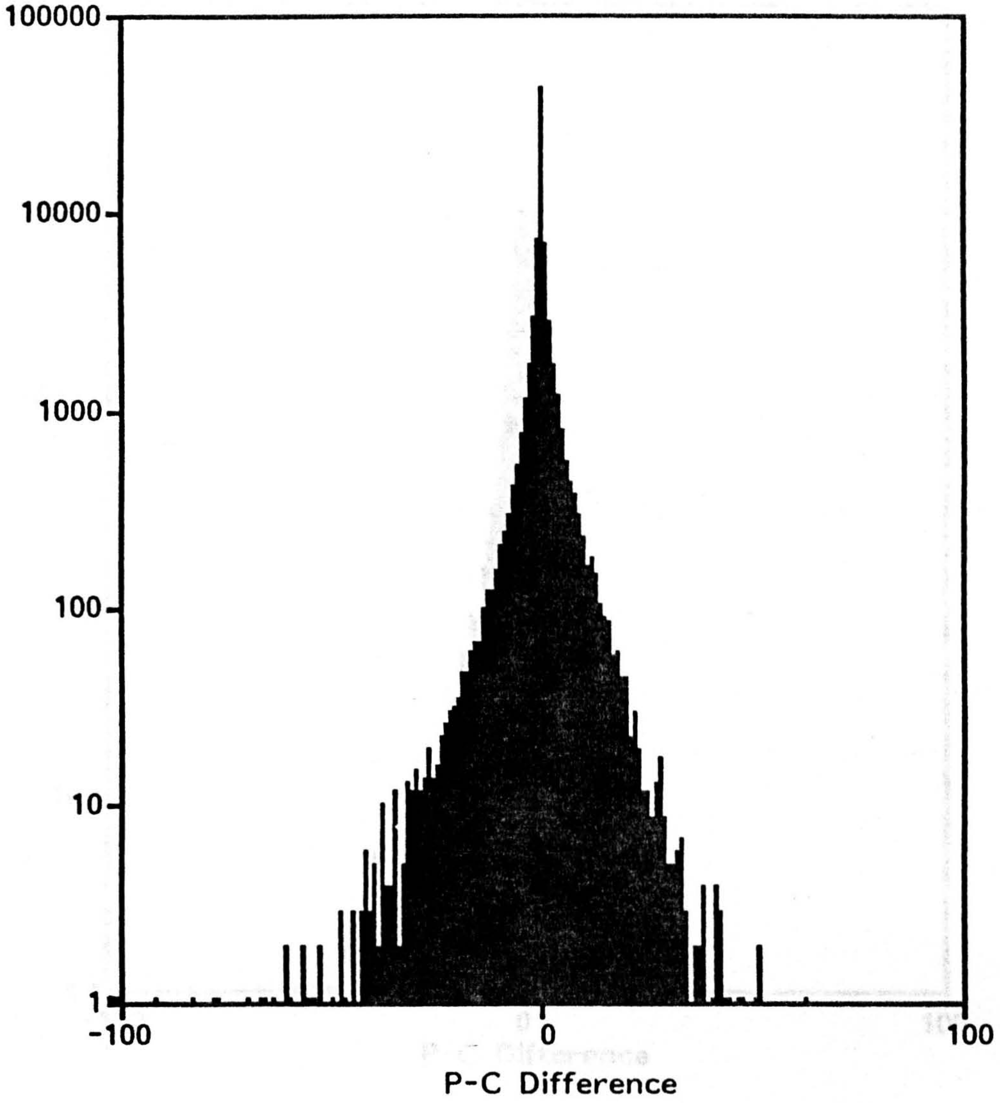


Figure 3.7. Distribution of Parent-Child Differences for all quad-trees constructed from a segmented frame from the test sequence "Miss America". The threshold c was 0.01. The vertical axis is log-scale. The horizontal axis is multiples of c.

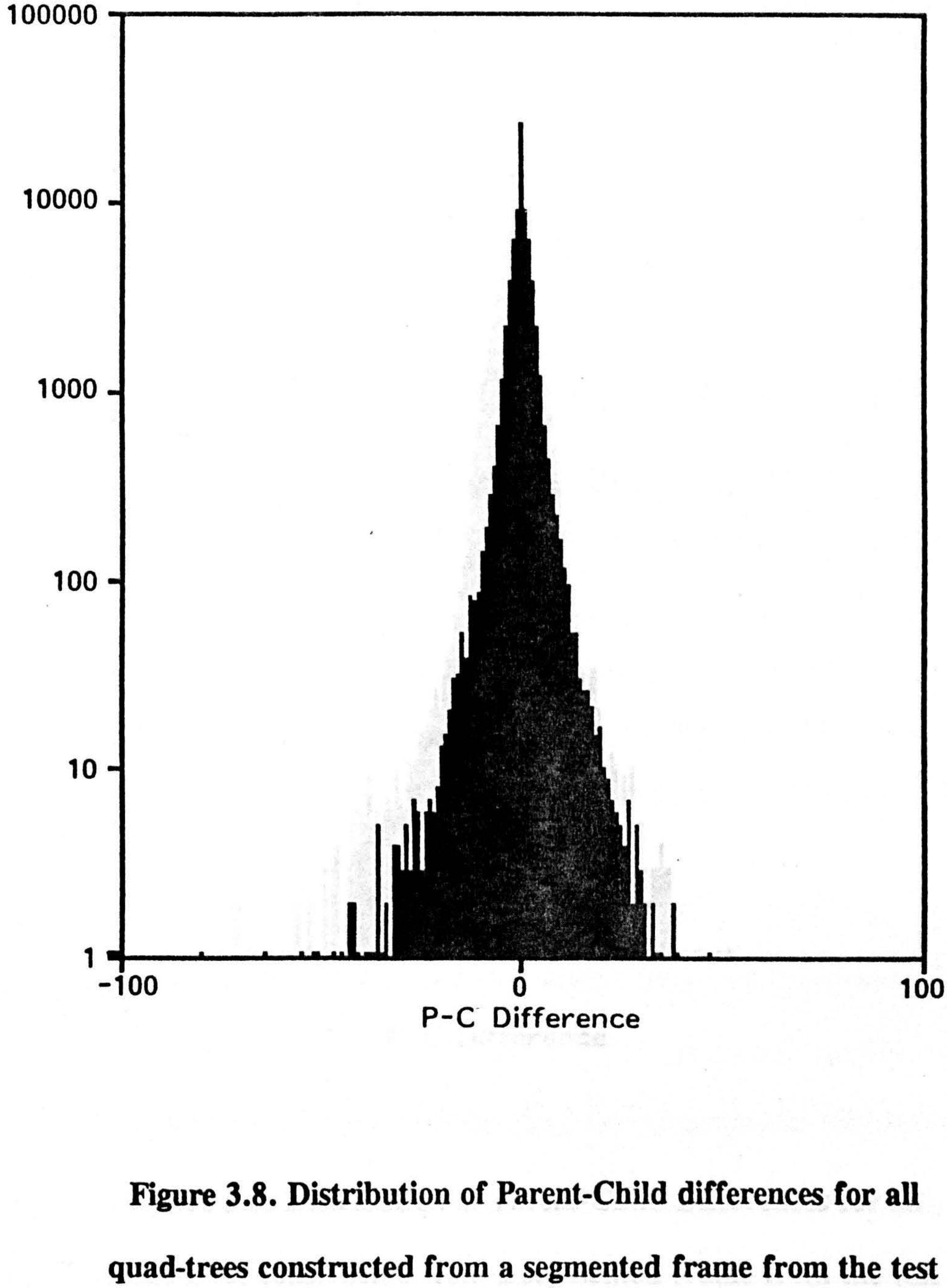


Figure 3.8. Distribution of Parent-Child differences for all quad-trees constructed from a segmented frame from the test sequence "Split Screen". The threshold c was 0.01. The vertical axis is log-scale. The horizontal axis is multiples of c.

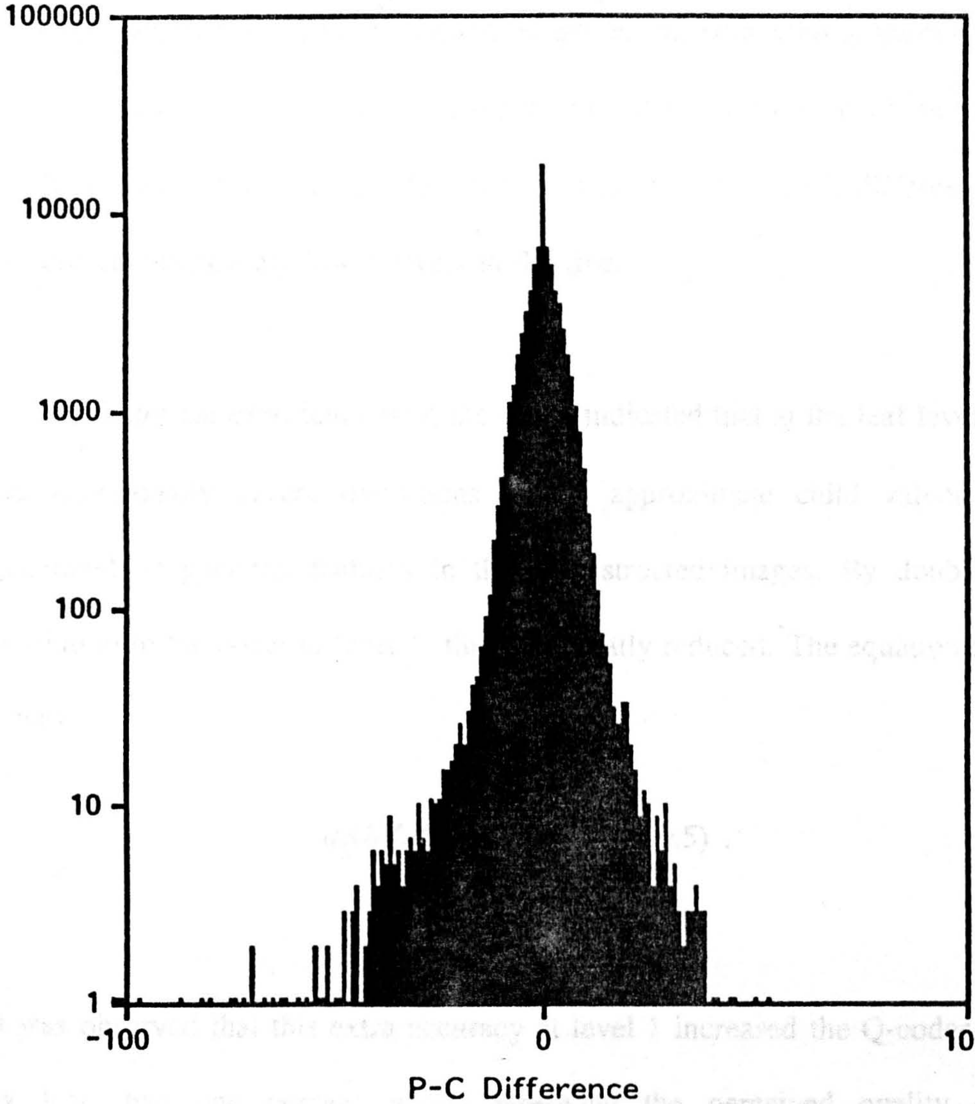


Figure 3.9. Distribution of Parent-Child Differences for all quad-trees constructed from a segmented frame from the test sequence "Trevor White". The threshold c was 0.01. The vertical axis is log-scale. The horizontal axis is multiples of c.

The integer difference value, which is actually a multiple of the Weber fraction, is generally in the range **-16 $\leq$ *q()* $\leq$ 16**, so a 5-bit quantiser is used as the structure of the context tree for the Q-coder. At lower levels in the tree, a 4-bit quantiser can be used effectively because the parent-child differences tend to zero at successively lower levels in the tree.

Empirical experience with the coder indicated that at the leaf level, there are occasionally severe distortions in the approximate child values which generated unappealing features in the reconstructed images. By doubling the resolution of the coder at level 1, this was greatly reduced. The equation at level 1 was

$$q_1(\delta Y, Y, c) = int(\frac{2 \cdot \delta Y}{c \cdot Y} + 0.5) \ . \qquad (3.9)$$

It was observed that this extra accuracy at level 1 increased the Q-coder output by less than one percent while improving the perceived quality of the reconstructed images. Note that occasionally the value sent to the 4- or 5-bit quantiser will actually exceed the range allowed and a flag of some sort must be sent to the Q-coder. By providing a separate context for this eventuality, which rarely occurs, the compression achieved generates very few bits. It is restricted by how small the Q_e estimate in the Q-code is allowed to go. This behaviour also occurs at the beginning of the region coding, in which the root itself is checked to see if it is representative of the entire tree.

Traditionally, extra bits must be sent to the decoder to indicate when the tree traversal is complete, i.e. when a node meets the criteria described above. However, the ability of the Q-coder to follow local entropy means that instead of signalling the termination of the traversal, a zero difference value can be sent instead, leading to many more bits sent to the Q-coder, but fewer bits sent to the decoder, because the flagging mechanism may be completely removed from the algorithm. There is no need to signal that the edge of a partially complete quad-tree has been reached because both the decoder and the coder have access to the edge map and therefore to the region shapes.

3.12. Discussion of the Coding Process

The complete coding algorithm is shown in figure 3.10. It is apparent that the entire frame is processed before any data is sent to the Q-coder. This is necessary in order to create the segmentation, which drives the region coding and decoding. The segmentation algorithm involves a number of steps. The image is first filtered to ensure that it is differentiable. This is necessary because the initial edge pixels are extracted using the Canny edge detector [72]. Contours are then extracted, closed and thinned to provide a complete edge map enclosing discrete regions, which are then numbered in a defined order. This work was carried out by a team at the GEC Hirst Research Centre in Wembley. The references to this procedure can be found in [56] and Canny's algorithm is originally found in [55]. The shaded boxes in figure 3.10 therefore indicate steps which have been performed separately from the rest of the coding process developed in this chapter.

The colour compression, or rather the compression of the two chromatic signals, is considerable because only a single pair of chromaticities need be sent for each region. This technique of assuming the modulation of the luminance signal by a single chromaticity vector is a high resolution analogy to the "colour washing" process used to add colour to classic black-and-white films [73]. "Colour washing" is an apt term in some ways because in addition to describing the technique, it portrays the result, which tends to be a slightly "washed out" version of the original. What this means of course is that the saturation is decreased; colours all take on a more neutral appearance. This is not at all noticeable in general. By finding the lowest values and the range, the quantisation for this part can be 3- to 4-bits per ordinate per region. The shape of the distribution varies, but is often approximately Gaussian.

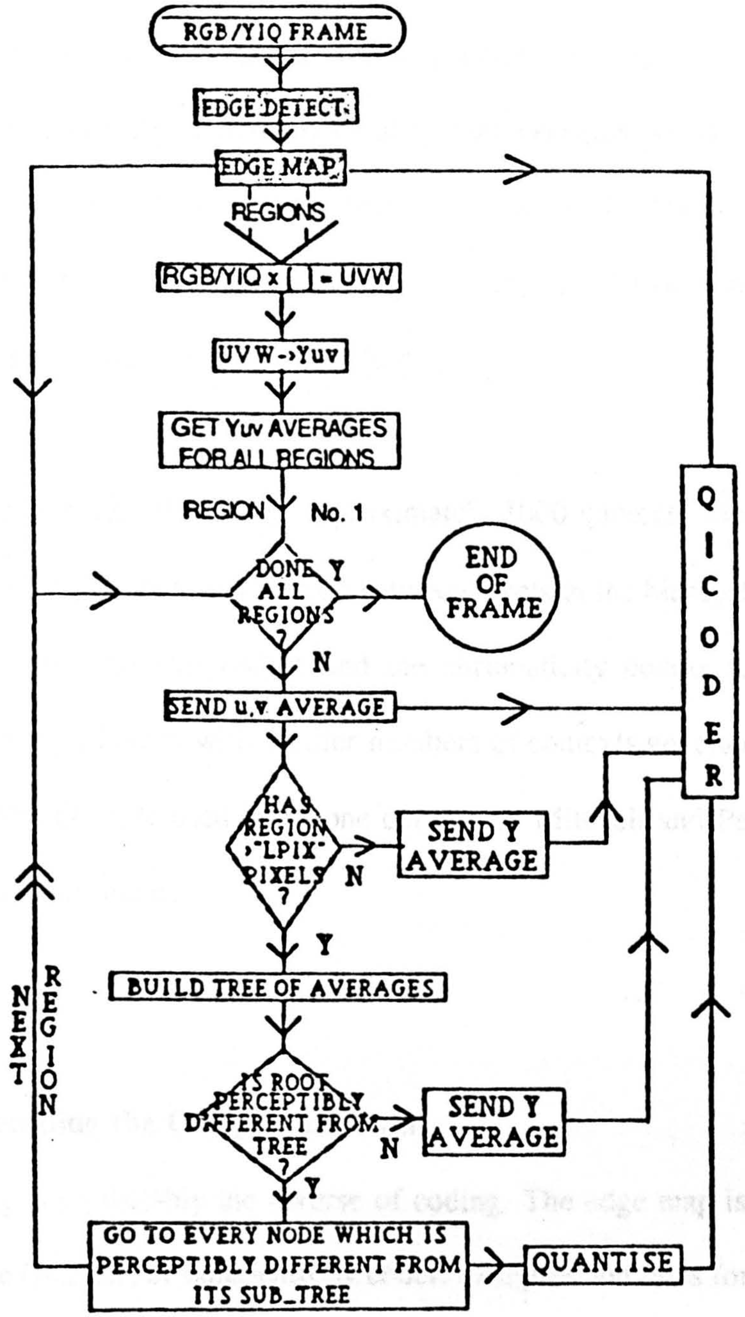


Figure 3.10. Flow diagram of the region coding algorithm. The shaded boxes marked "Edge Detect" and "Edge Map" indicate steps which were not performed by the algorithms developed as part of this thesis. (Source: McColl and Martin [61])

Rather than encode all regions with a quad-tree, all regions with fewer than "LPIX" pixels (see figure 3.10) have only their averages coded. Experiments indicate that this number should be around 16. The variation in reconstruction quality can be large, especially if there are a lot of small regions around an object of importance such as a face.

For the coder described here, approximately 1000 contexts were used to drive the Q-coder. The statistical variation between levels of the binary LPS/MPS tree for the 4- and 5-bit tree coders and the chromaticity coders tend to be pronounced, and experiments with smaller numbers of contexts generated poorer compressions. The Q_e table used is the one derived by Mitchell and Pennebaker for their DPCM experiments.

3.13. Decoding the Compressed Data

Decoding is predictably the reverse of coding. The edge map is received initially from the Q-coder, or other entropy coder. Compression rates for the edge map are on the order of 0.1 bpp [74] (i.e. 0.1 x 65536 pels), and there are usually 8000-10000 edge pixels. The decoder can then construct the region layout and build quad-trees when necessary. If the decoder detects that a given region is sufficiently energetic to need a tree traversal to reconstruct the pixels, then the nodes are visited in turn, and the predicted value is calculated from the parent and the quantised difference. Because all parent-child differences are transmitted, even zeros, in order to exploit the Q-coder behaviour, then the tree traversal cannot stop when a zero difference is received, otherwise the synchronicity between coder and decoder may be lost. This means that extra compression is gained at the expense of a longer decoding cycle per region.

Once the luminance data is reconstructed from the decoder and the (u_r, v_r) relative chromaticity offsets are added to the local "origin", the reconstructed $(UVW)_q$ triplets are found from simple equations similar to (2.5) and (2.6), i.e.

$$(U+V+W)_q = \frac{255 \cdot Y_q}{v_q} , \tag{3.10a}$$

$$U_q = \frac{v_q \cdot (U+V+W)_q}{255} , \tag{3.10b}$$

$$V_q = Y_q , \tag{3.10c}$$

and

$$W_q = \frac{(255-u_q-v_q) \cdot (U+V+W)_q}{255} . \tag{3.10d}$$

and the equivalent $(RGB)_q$ vectors can be calculated by matrix multiplication.

3.14. Edge Reconstruction

Now that the region data has been reconstructed, there remains the task of choosing suitable pixel values for the edge pixels, whose values are not transmitted. Typical schemes involve choosing the average of the pixels around the edge pixel [56], or alternatively generating pixels which emphasise an effect such as seen in Mach bands, in order to make the resulting image seem adequately sharp. Consideration can also be given to the use of an operator which behaves differently depending on the region activity, so that reconstructed data such as faces do not end up becoming "blocky", as may occur with simple averaging. In practice, however, the results of using a simple averaging were not perceptibly different from more time-consuming algorithms, so this scheme was used. Often, random noise is added to the reconstructed image after this stage in order to dither the image and thus break up any severe contouring introduced by the edge synthesis. Because of the interest in perceptual distortion and its parameters, postprocessing by noise addition has not been performed in this study.

3.15. Smoothing of Quad-Tree Aliasing

One important drawback of using quad-trees for region reconstruction is the aliasing introduced by the upward averaging process. After reconstruction, this aliasing is seen as a blockiness in the image, with the block size dependent on the level in the tree at which the greatest aliasing occurred. There are two ways to overcome this problem, both based on Clippingdale's method [63] of adding extra nodes to the quad-tree structure. These nodes can be called "phantoms" because they do not form a quad-tree themselves but rather straddle the boundaries between the artifactual "blocks" in the reconstructed images. The number of extra nodes in the tree is given from the following. At level l in the tree $(1 \leq l \leq n-1)$, in which there are, from before, $2^{(n-l)2}$ original nodes, there are $(2^{n-l}-1)^2$ phantom nodes added; these lie between four original nodes, as shown in figure 3.11, which also illustrates the upward averaging process described in the next paragraph. The children of these nodes are simply the four original nodes closest vertically to this one at the next lowest level in the tree. Rather than having a single parent, though, these children instead have four different original parents.

The upward averaging process involves assigning a phantom node the average of its children in the normal fashion (see figure 3.11). Proceeding from the root down the tree to the leaves, then, the assignment of the children is modified, for the simple case of a complete tree, to

$$s_{i,j,k} = \frac{\bar{s}_{\frac{i}{2},\frac{j}{2},k} + \bar{p}_{\frac{i}{2},\frac{j}{2},k}}{2} \, . \tag{3.11}$$

Because the tree may not be complete, this equation is modified to a form similar to (3.2), i.e.

$$s_{i,j,k} = \frac{\bar{s}_{\frac{i}{2},\frac{j}{2},k+1} \cdot t_{\frac{i}{2},\frac{j}{2},k+1} + \bar{p}_{\frac{i}{2},\frac{j}{2},k+1} \cdot q_{\frac{i}{2},\frac{j}{2},k+1}}{t_{\frac{i}{2},\frac{j}{2},k+1} + q_{\frac{i}{2},\frac{j}{2},k+1}}, \tag{3.12}$$

where *q* represents leaf coverage in a similar manner to *t*. Figure 3.12 illustrates the interpolation process on descent of the quad-tree.

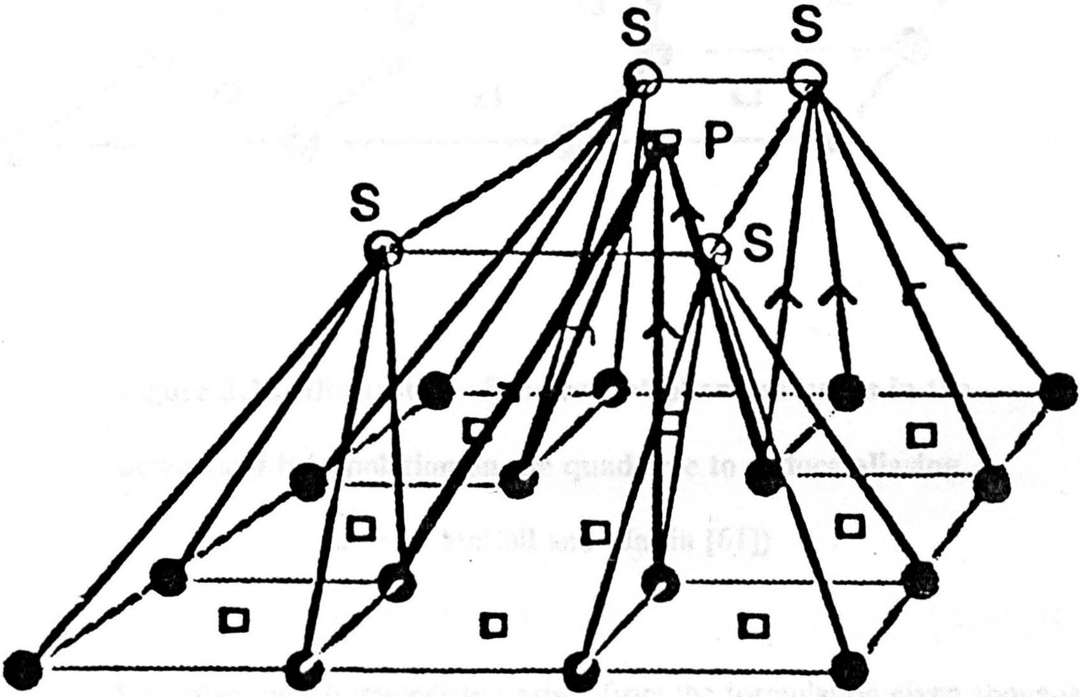


Figure 3.11. Illustration of phantom nodes in the quad-tree and their assignment in the upward averaging process.

(Source: McColl and Martin [61])

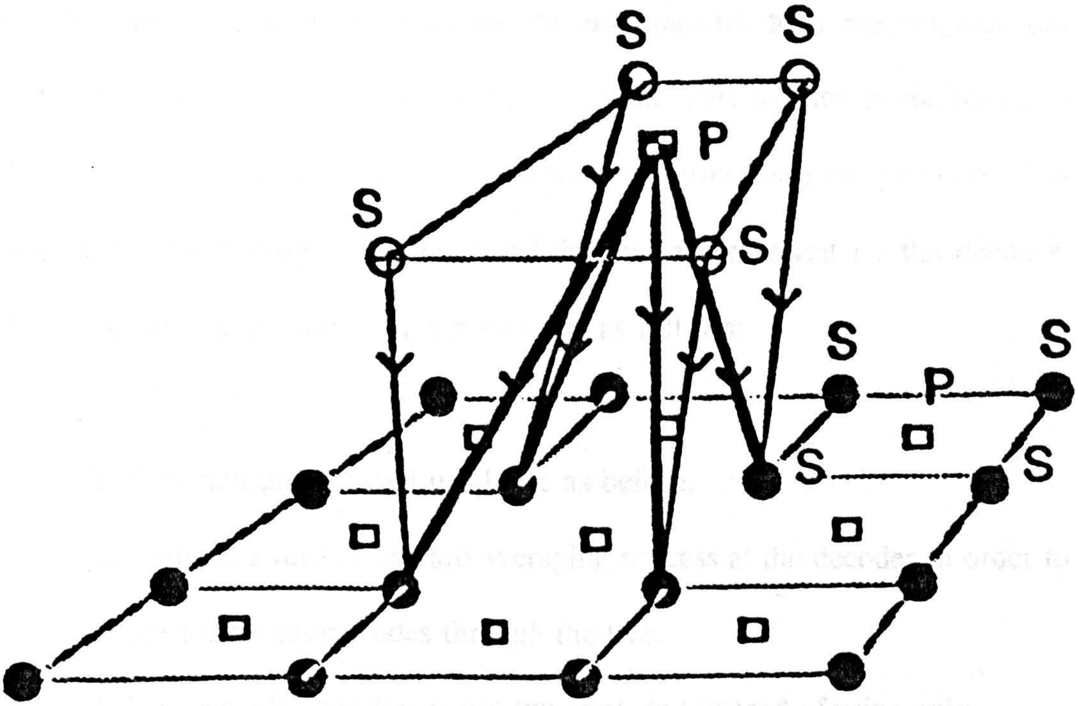


Figure 3.12. Illustration of the role of phantom nodes in the downward interpolation on the quad-tree to reduce aliasing.

(Source: McColl and Martin [61])

A problem which immediately arises from the formulation given above is the situation in which either the number of nodes covered by the original nodes is much smaller than that covered by the phantom, or vice versa. This situation can often occur at nodes near region perimeters. In this situation, the best solution has been to adopt the use of (3.11).

The two solutions involve the use of the phantom nodes in the coder and decoder. The first solution is to encode and transmit both the original and phantom nodes by modifying the coding algorithm. This solution is inadequate to remove the blocking effect however, because the effect is more pronounced in regions for which only a few levels of the tree are received by the decoder. Hence the second solution. The algorithm is as follows:

1. Generate the decoded quad-tree as before.

2. Perform a further upward averaging process at the decoder in order to create the phantom nodes through the tree.

3. Perform a further downward traversal, but instead of using only original nodes to generate the children, use (3.12).

The result of using this simple modification, which requires no transmission overhead at the coder, is a considerable reduction of the block artifacts. Often they are not visible in any way. It should be noted however, that this is simply a smoothing operation to remove an objectionable artefact; as such, extra noise is in fact being introduced to the reconstruction process, in a way that leads to a more appealing result to the observer.

3.16. Test Results with Varying Compression and Reconstruction Quality

In the test images that follow, the parameter used to control the coding process was simply a constant, the "just noticeable contrast difference" threshold, *c*. The value was adjusted in order to reduce the number of transmitted bits or conversely increase the reconstruction quality. A discussion of this parameter will follow the results. The complete decoder flow diagram is shown in figure 3.13.

Figures 3.14. 3.15 and 3.16 show the original test images used to drive the region coding algorithm. The edge maps corresponding to these images are shown in figures 3.17, 3.18 and 3.19. These maps were generated by the GEC research group [56]. Table 3.1 provides the various statistics generated as the parameter *c* was adjusted. In column five of the table, the "Bit Rate" value reflects the compression achieved from coding the region data alone, by dividing the "Q-Coder Bits Out" values by 65536. The true transmission cost for the whole frame can be approximated by adding 0.1 or so to this value.

The experimental conditions under which all the reconstructions were viewed were the same as those used in the previous chapter. These are described in appendix A.

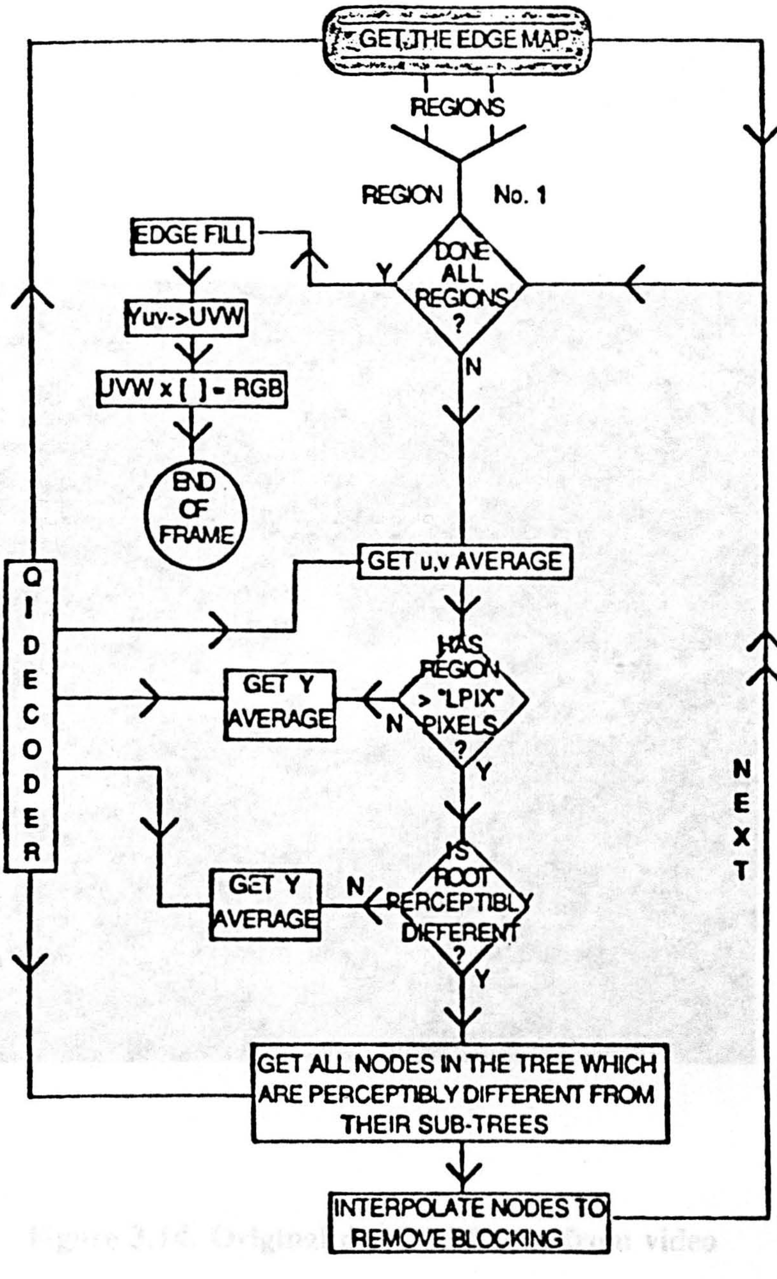

GET THE EDGE MAP
REGIONS
REGION No. 1
DONE ALL REGIONS ?
Y
N
EDGE FILL
Yuv->UVW
UVW x [ ] = RGB
END OF FRAME
GET u,v AVERAGE
HAS REGION > "LPIX" PIXELS ?
N
Y
GET Y AVERAGE
IS ROOT PERCEPTIBLY DIFFERENT ?
N
Y
GET Y AVERAGE
QI DECODER
GET ALL NODES IN THE TREE WHICH ARE PERCEPTIBLY DIFFERENT FROM THEIR SUB-TREES
INTERPOLATE NODES TO REMOVE BLOCKING
NEXT REGION

Figure 3.13. Flow diagram for the decoder algorithm. The shaded box marked "GET THE EDGE MAP" is not included in the decoder implemented as part of this thesis.

(Source: McColl and Martin [61])

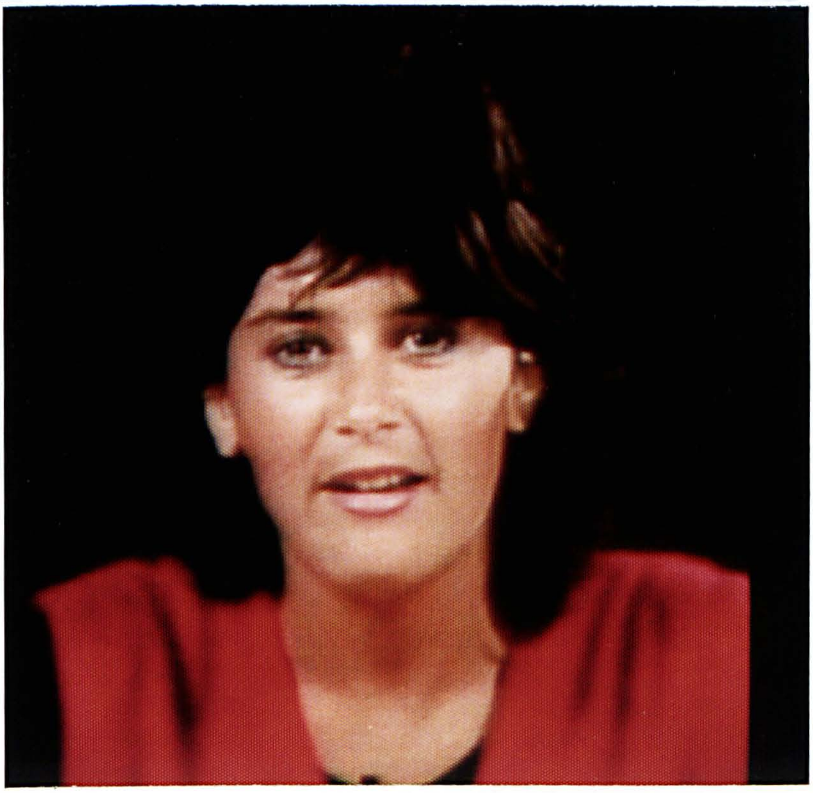


Figure 3.14. Original digitised frame from video codec test sequence "Miss America" used to generate edge maps and region data for codec algorithms.

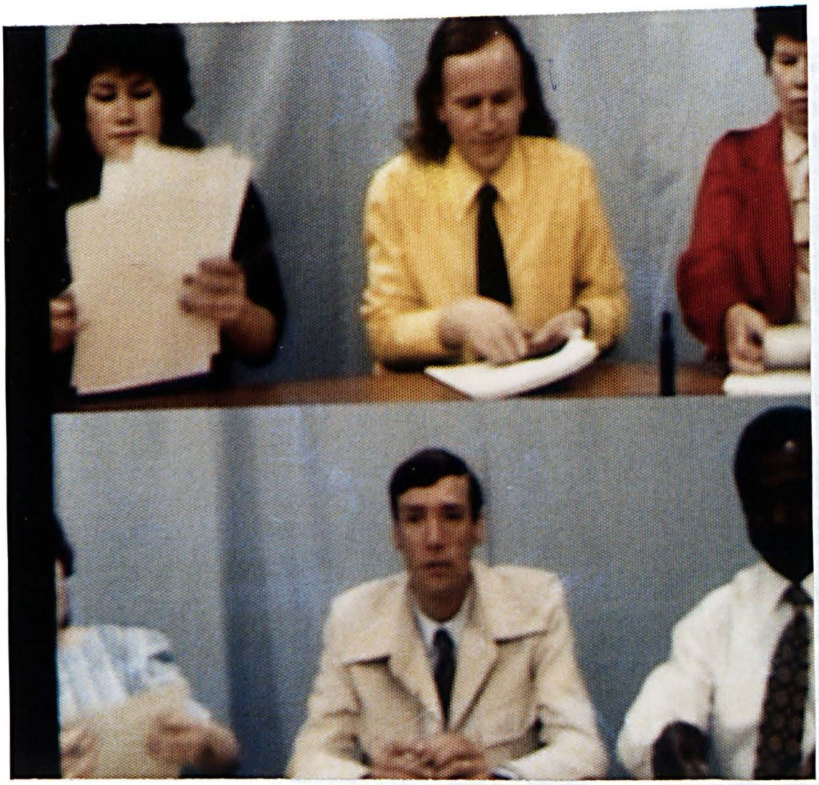


Figure 3.15. Original digitised frame from video codec test sequence "Split Screen" used to generate edge maps and region data for codec algorithms.

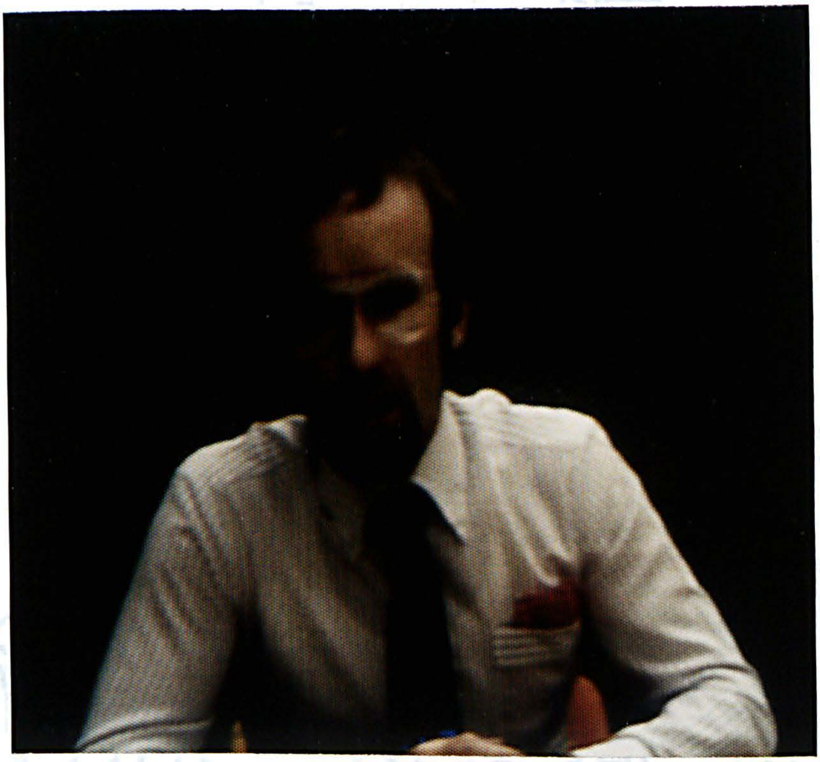


Figure 3.16. Original digitised frame from video codec test sequence "Trevor White" used to generate edge maps and region data for codec algorithms.

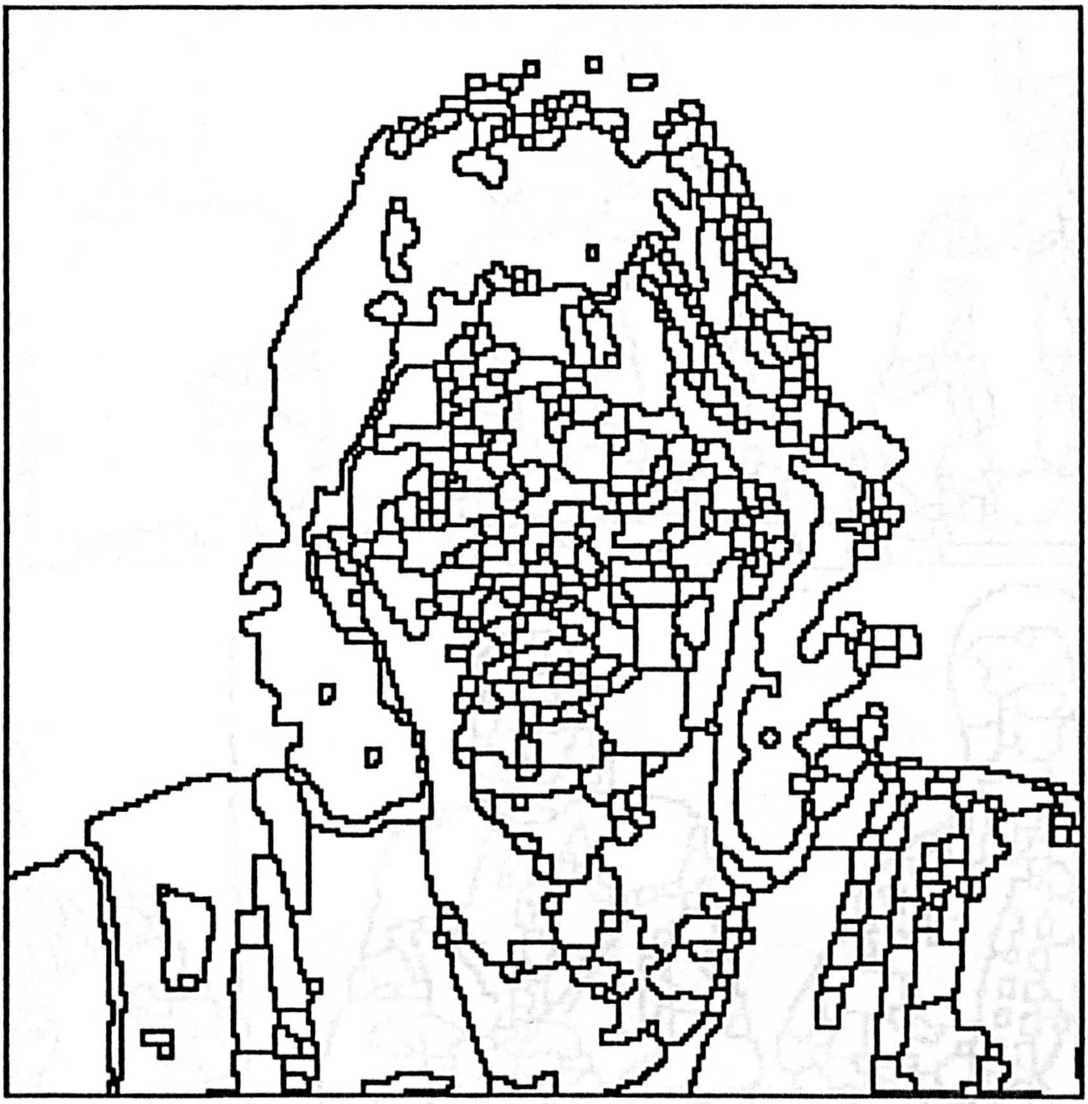


Figure 3.17. Edge map computed for frame from test sequence "Miss America", for use in region coding. Corresponding original frame is shown in figure 3.14.

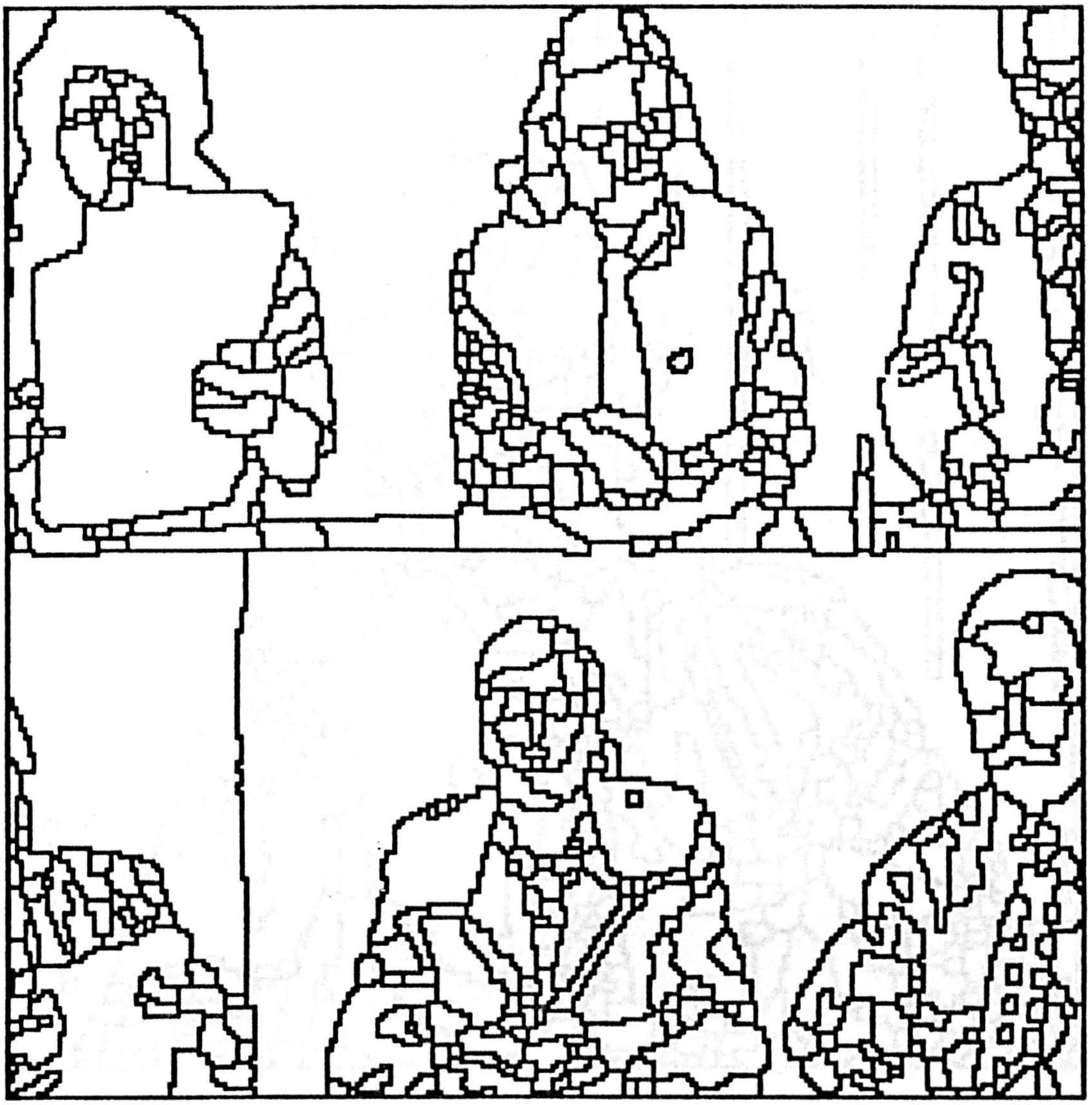


Figure 3.18. Edge map computed for frame from test sequence "Split Screen", for use in region coding. Corresponding original frame is shown in figure 3.15.

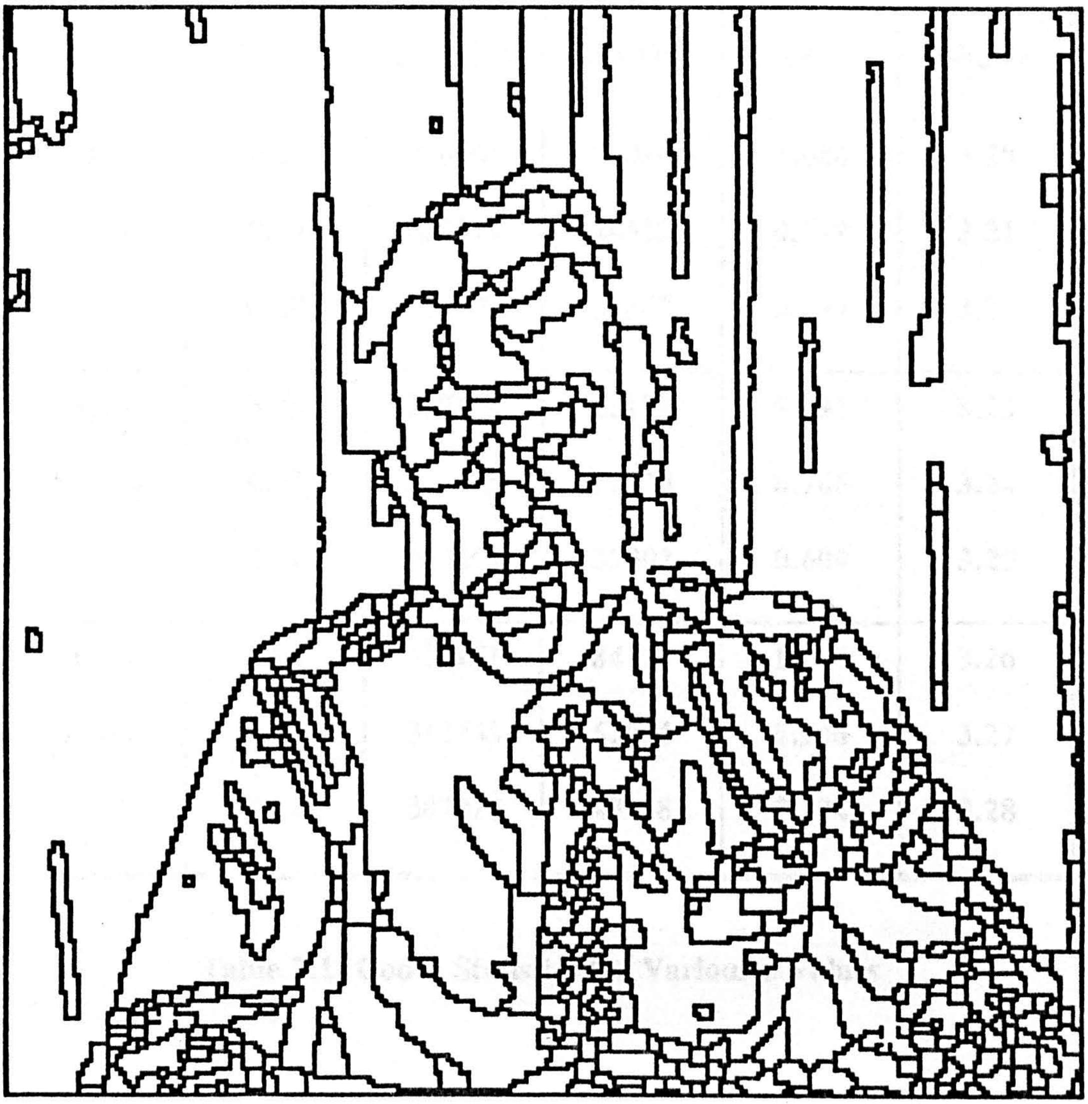


Figure 3.19. Edge map computed for frame from test sequence "Trevor White", for use in region coding. Corresponding original frame is shown in figure 3.16.

Image Sequence	c Value	Raw Bits Out	Q-Coder Bits Out	Bit Rate (bpp)	Figure Shown
Miss America	0.06	330046	67090	1.024	3.20
	0.08	322919	44512	0.679	3.21
	0.10	318730	32347	0.494	3.22
Split Screen	0.06	327156	65194	0.995	3.23
	0.08	323445	50224	0.766	3.24
	0.10	318607	39902	0.609	3.25
Trevor White	0.08	322171	84114	1.284	3.26
	0.10	312738	62014	0.946	3.27
	0.12	307028	48108	0.734	3.28

Table 3.1. Coder Statistics for Various c Values

For each of the entries in table 3.1 there is a corresponding reconstruction. For the "Miss America" and "Split Screen" sequences, the reconstructions, for *c* values of 0.06. 0.08 and 0.10, are displayed in figures 3.20, 3.21, 3.22 ("Miss America"), 3.23, 3.24 and 3.25 ("Split Screen") respectively. For the "Trevor White" sequence, the reconstructions using c values of 0.08, 0.10 and 0.12 are shown in figures 3.26, 3.27 and 3.28 respectively.

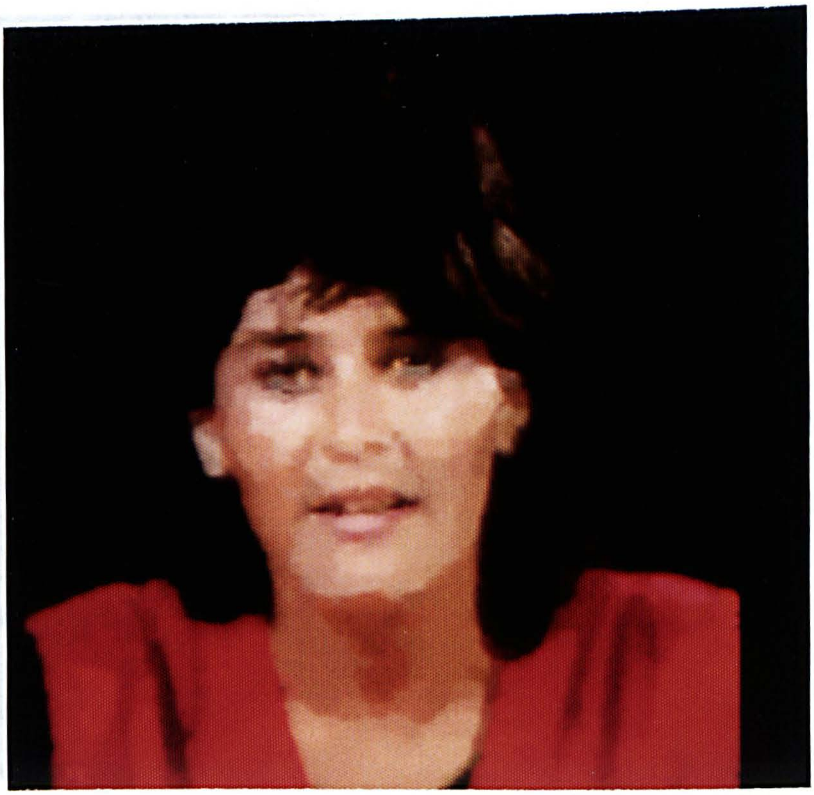


Figure 3.20. Reconstructed frame from test sequence "Miss America".

The contrast threshold value for this reconstruction was c=0.06.

The total bit rate for all regions was 1.024 bpp.

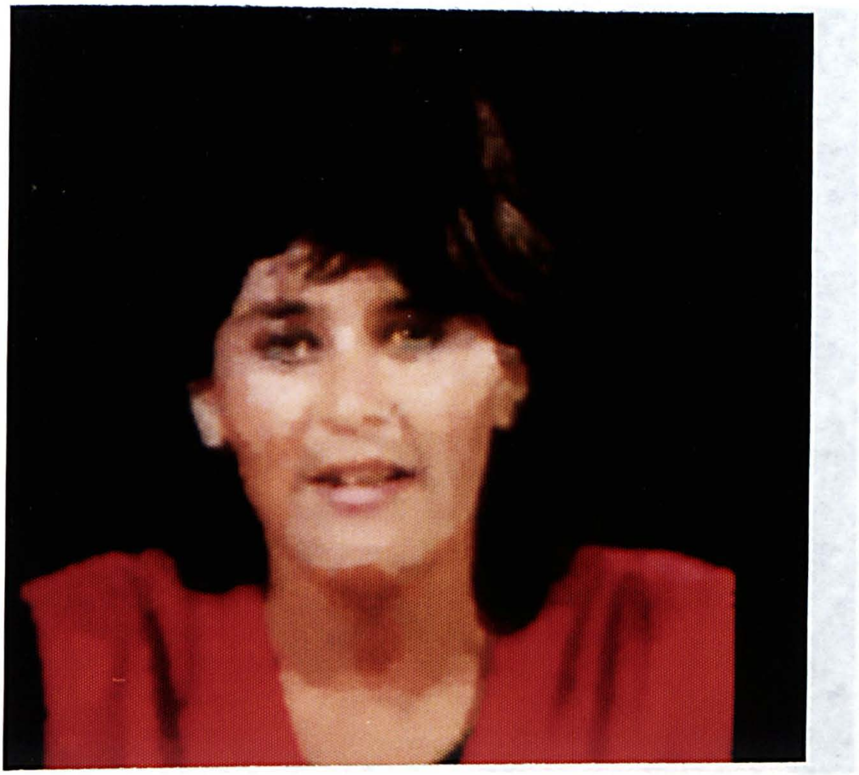


Figure 3.21. Reconstructed frame from test sequence "Miss America".

The contrast threshold value for this reconstruction was c=0.08.

The total bit rate for all regions was 0.679 bpp.

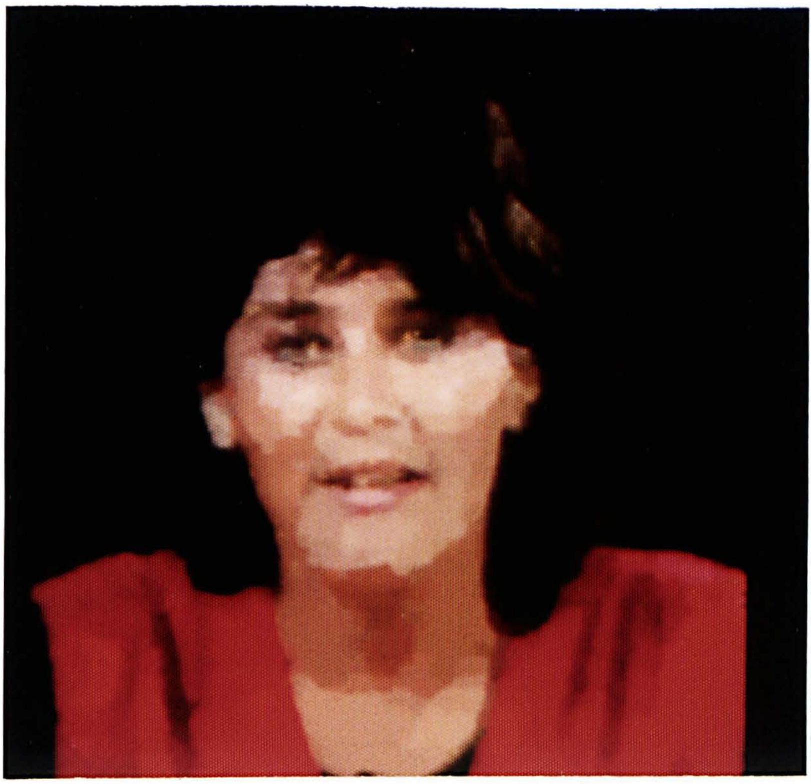


Figure 3.22. Reconstructed frame from test sequence "Miss America".

The contrast threshold value for this reconstruction was c=0.10.

The total bit rate for all regions was 0.494 bpp.

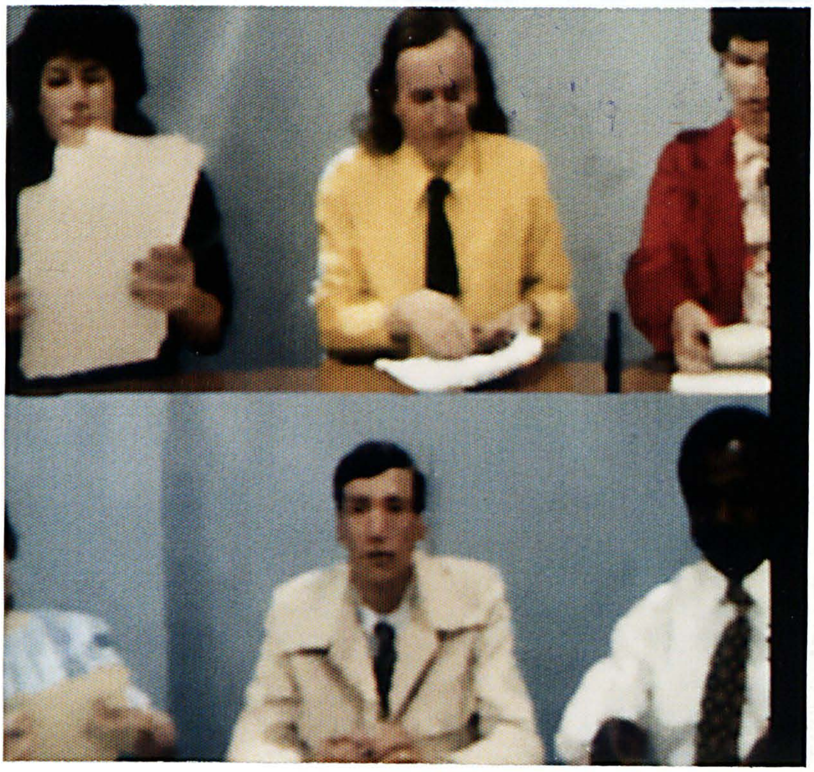


Figure 3.23. Reconstructed frame from test sequence "Split Screen".

The contrast threshold value for this reconstruction was c=0.06.

The total bit rate for all region pixels was 0.995 bpp.

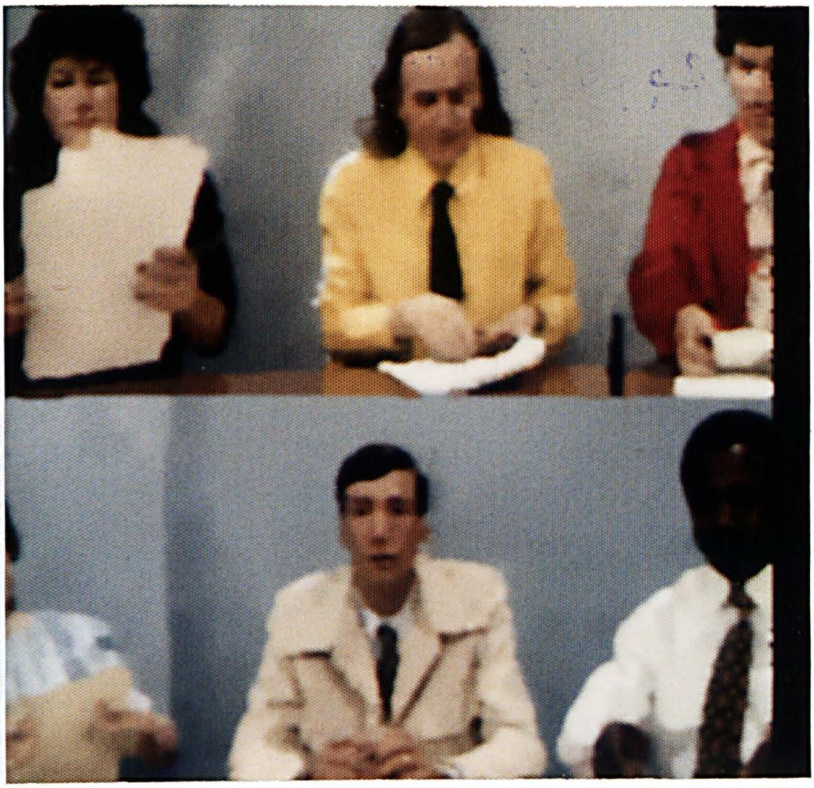


Figure 3.24. Reconstructed frame from test sequence "Split Screen".

The contrast threshold value for this reconstruction was c=0.08.

The total bit rate for all region pixels was 0.766 bpp.

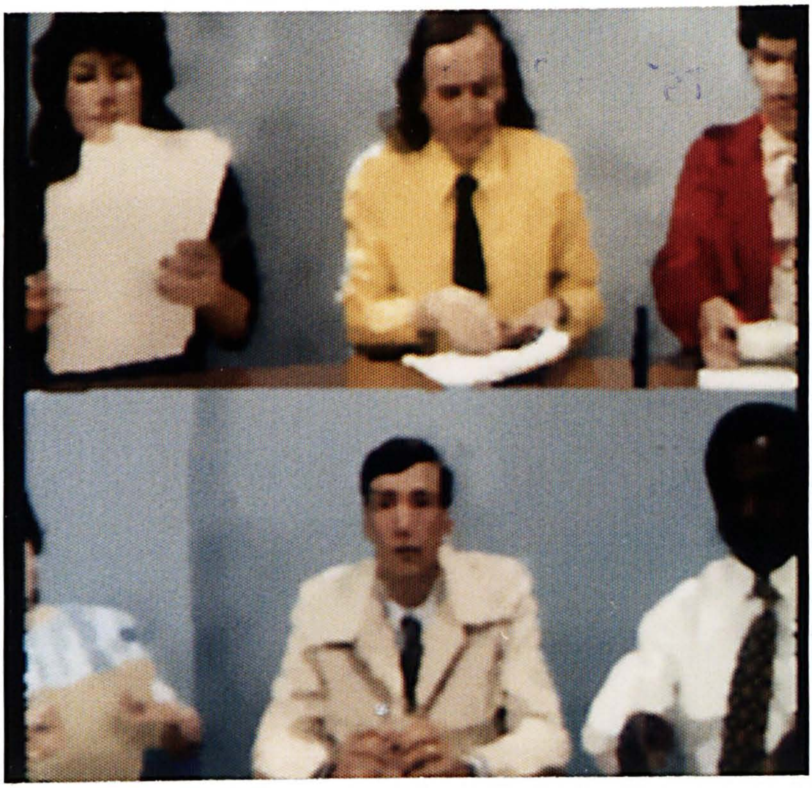


Figure 3.25. Reconstructed frame from test sequence "Split Screen".

The contrast threshold value for this reconstruction was c=0.10.

The total bit rate for all region pixels was 0.609 bpp.

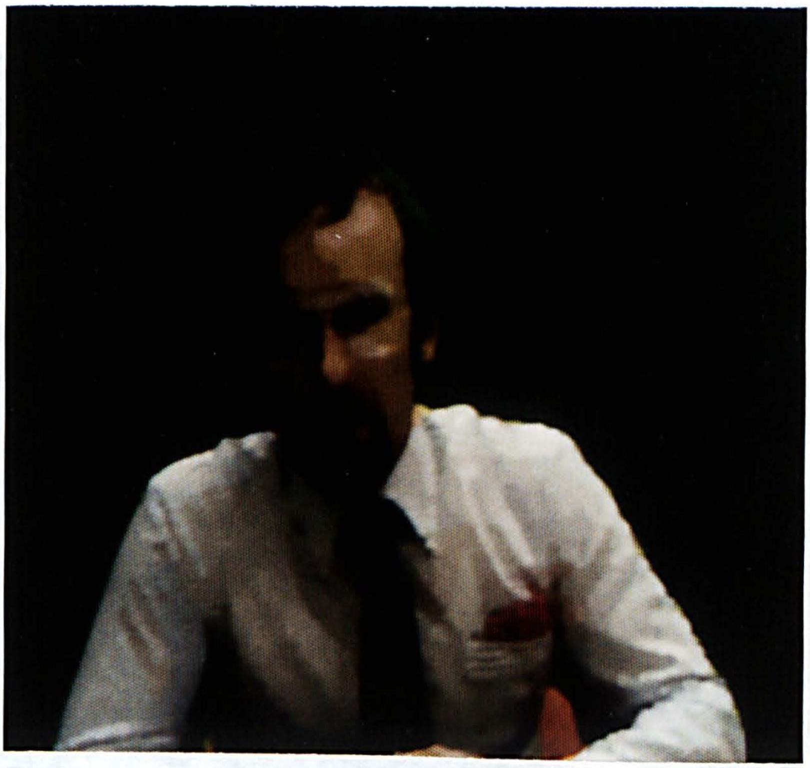


Figure 3.26. Reconstructed frame from test sequence "Trevor White".

The contrast threshold value for this reconstruction was c=0.08.

The total bit rate for all region pixels was 1.284 bpp.

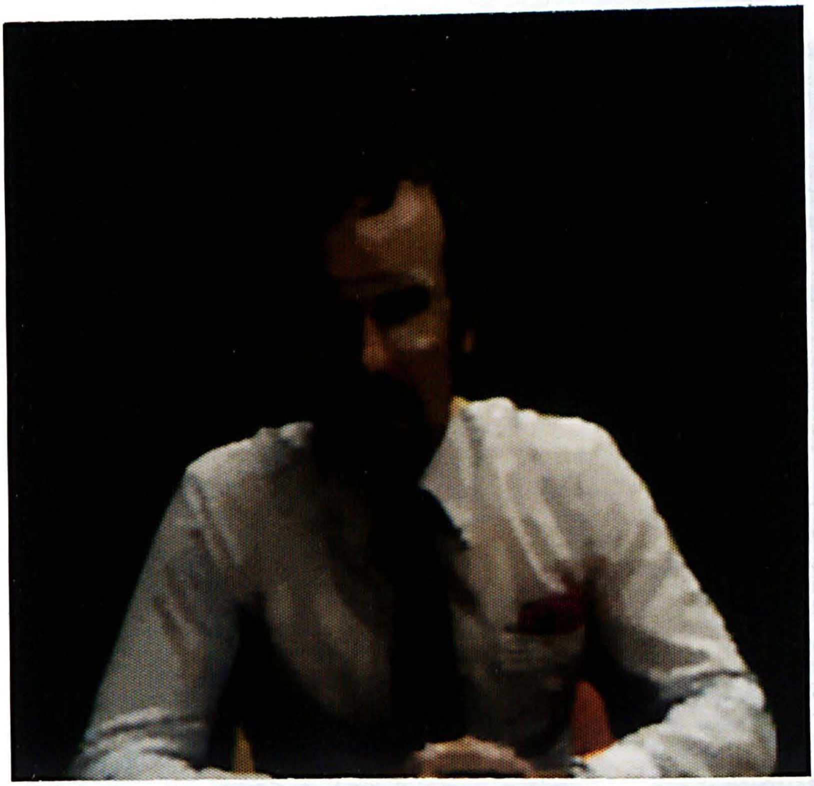


Figure 3.27. Reconstructed frame from test sequence "Trevor White".

The contrast threshold value for this reconstruction was c=0.10.

The total bit rate for all region pixels was 0.946 bpp.

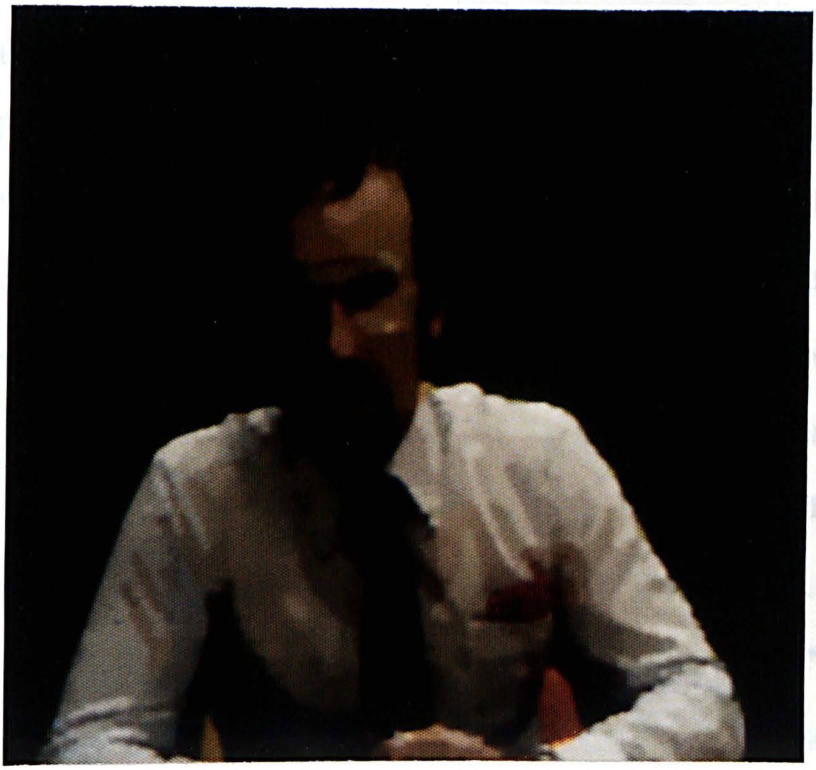


Figure 3.28. Reconstructed frame from test sequence "Trevor White".

The contrast threshold value for this reconstruction was c=0.12.

The total bit rate for all region pixels was 0.734 bpp.

It is apparent that the increase in bits required to adequately reproduce the "Trevor White" frame reflects the detail present. From the data, it is also apparent that for the results of this study, a figure for *c* of less than 0.08 seems appropriate in order to reproduce all the detail in the "Trevor White" frame. By contrast, the "Miss America" frame is adequately reconstructed with a much higher value. It is interesting to note that when comparing the results obtained from the "Miss America" frame and the "Split Screen" frame, coding the former at "low" *c* values generates more bits than does the latter, and vice versa at "high" *c* values. This behaviour is apparent when the various regions in each frame are studied. The activity in the large number of small regions in the face of "Miss America" is such that as the contrast threshold ratio is reduced, the number of bits generated rises dramatically. As the threshold increases, however, the large regions representing the curtain backdrop in "Split Screen" begin to generate a large number of pixels relative to the output from "Miss America". The behaviour shown in the table can be considered to be a reflection of the different perceptually oriented features in the coding-decoding algorithm.

Although the "Miss America" results provided the highest compression factors, the large portion of the image occupied by the model's face, which is segmented into many small regions, leads to a blurred reconstruction at each threshold value. By contrast, the "Trevor White" results are much more "in focus", due to the smaller number of regions which comprise the model's face.

The impact of the segmentation should be considered when discussing the results; we cannot consider one component to the exclusion of the other.

The results presented here are, I feel, a definite improvement over those found in other comparable literature, such as [51] and [52]. No other schemes which attempt to encode colour region data in a two-component model have been found in the literature, unfortunately, which limits the sort of claims one can make about the success of this method.

3.17. Summary

The use of a simple perceptually oriented parameter in a region-coding model has been demonstrated. The results vary in appeal and compression, according to image complexity. Because of this variation, the implication is that this parameter is either insufficient or improperly applied. In fact, as mentioned earlier, the visual system has a varying contrast sensitivity in relation to the spatial frequency of the signal being observed. At the very least this means that the value of c should be increased toward the bottom of the quad-tree to take into account the likely increase in the rate of variation between averaged pixels. The incorporation of this minor modification can be easily incorporated into the algorithm; the "perceptual difference" value would be modified to a level-dependent value $c=c(l)$. A further incorporation which would not be difficult to add is the "bath tub curve" that the Weber fraction traces; at high and low intensities, the nominal value increases considerably (see Stevens [45]). What is ideally required however is a measure of spatial frequencies in the given sub-tree, which would then modify *c* accordingly. An isotropic solution to this would be to find a spatial-frequency transform of the data and apply a threshold function to determine what energy will be lost at a given radial frequency. Unfortunately this procedure is impractical due to the convoluted shape of the region, which effectively imposes a severe windowing effect on the resulting spectrum. This effect will be discussed further in the next chapter. Further improvement in compression and image quality may also be gained by considering monitor gamma and its interaction with the visual system and incorporating these factors into the expression for the threshold parameter. In chapter two, a linear relationship seemed to be occurring. It is likely to be more complex than that, but the net effect may be that the threshold parameter can be better expressed as a combination of $\frac{\Delta Y}{Y}$ and ΔY.

Chapter 4. Textures - Analysis and Synthesis

Part of this study involved an investigation of image texture, with a view to synthesising all or part of a colour image from some succinct description of the image derived from an analysis of the textures found in the image. Given that, from before, we have a set of images which have already been segmented in some way, this chapter is concerned with the possibility of extracting and transmitting some simple repeating pattern or set of parameters which can be used to synthesise the regions at the receiver. There has been much research into the understanding, measurement and manipulation of textures. Despite all this work, the definition of texture is rather tenuous, being more easily described in terms of its functional properties than as an object or characteristic. Texture is defined by Faugeras [75] as a "region property or feature of an image that characterizes the structural relationship within the region." Others are more able to describe what texture analysis can be used to achieve rather than what texture is *per se*. Hence texture analysis is used to classify images, to recognise a particular feature in an image, and of course to segment the image. Finally, what is of interest to this study is the 'filling in' of these textured areas in the reconstructed image by performing a synthesis of the original texture in such a way as to be visually pleasing (or at least acceptable).

This chapter involves the development of an algorithm which is designed to synthesise, in an fashion which leads to an appealing result, regions within the test sequences described before in which there is an identifiable texture. Particular emphasis is placed on the successes of two recent investigations found in the literature, and some comparisons are made between the compression and image quality achieved with the algorithm developed in this chapter, and that achieved by the algorithm in the previous chapter for similar parts of the example images.

4.1. Texture Models - Statistical and Structural

There are two main approaches to texture analysis. Davis [76] has argued that there are in fact three, but in his survey, the second and third categories could be considered as sub-categories with a common ancestor. These are not presented in order of importance. Most of the early models of texture do fall into the first category so the order may be considered roughly chronological.

4.1.1. Statistical Models

Statistical models of texture involve the identification of global statistics such as moments (mean and variance); correlation measures are also used to provide a description of the texture. The use of Markov chains and autoregressive functions are used to provide a suitable "fit" for the texture from the original.

Julesz [77-79] has contributed many papers in this area. He reported [77] that human subjects were unable to discriminate between synthetic textures with equal first- and second-order statistics or moments, but differing higher-order statistics. Counter examples have since proved that this is not always so, however. Julesz more recently reported that in fact local features within the texture, which he named *textons* [79], were the basis for texture discrimination and that differences in the first-order statistics of the textons themselves was a sufficient condition for discrimination. Voorhees and Poggio [80] have reported an algorithm to identify and use such elements for segmentation of real images. There are few examples of such applications, however.

The importance of second-order statistics of the texture density function implies that the autocorrelation function, which is the expectation of the second-order density function of the texture, is a useful texture-measurement function. The use of the power spectrum, which is in fact the Fourier transform of the autocorrelation, is also used extensively in the literature. Pratt [81] notes however that no single statistical function on its own is sufficient as a textural descriptor. He also argues that Julesz' original conjecture holds in the main.

The Gray Level Co-occurrence Matrix (GLCM) (see Davis [76], Rosenfeld and Kak [82]) has been widely studied by Haralick as a texture discriminator and descriptor [83-85]. He has also called this the Gray-Tone

Spatial Dependence Matrix. The GLCM is a joint histogram of gray-scales which is an analogy to a correlation, emphasising patterns of gray-level variations about the diagonal axis of the local image. The dimensions of this structure grow with the size of the texture area and with the size of gray scale unfortunately, which limits its practicality in real images. Texture segmentation algorithms using the GLCM over windows in an image have been reported however [86]. There are many other statistical techniques which use similar devices to find a "direction" in which a given texture sample "lies" i.e. the orientation of the texture [76,87,88].

Another statistical approach to texture analysis is to model the texture as the result of filtering a random process [75]. Decorrelating the texture using "whitening" filters leads to an estimation of the generating process, and hence to the filter function. The whitening is performed on the autocorrelation function of the texture, or rather its Fourier transform, i.e. the power spectrum.

Gagalowicz [89] has used the colour vision model developed by Faugeras [29] to generate colour texture fields using the same technique as that of Faugeras and Pratt and has shown that, for the Faugeras model, texture fields with the same second order statistics are indistinguishable, regardless of which primary is tested. There are no other colour texture analyses that the author has found in the literature.

4.1.2. Structural Models

The structural approach to texture discrimination has been studied since the late 1960s. Hawkins [90] was one of the first to view texture as a regular or semi-regular placement of a 'primitive' over a whole region, so that the reconstruction resembles a mosaic, or tiling. The role of the texture primitive in texture analysis, description and also in texture synthesis has been researched extensively. Zucker [91,92] conceived of an ideal texture, composed of a single primitive, regularly repeated over the whole field; geometric transformations to each instance of the primitive could then be used to synthesise a "surface texture" as an approximation to the original. Rosenfeld *et al.* [93,94] have studied texture extraction and discrimination using various methods; they have studied various placement rules to provide for realistic synthesis. Gonzalez and Wintz [95] contains a discussion of placement rules and the statistics used to derive them. Ahuja and Rosenfeld [93] proposed the concept of hierarchical textures, in which a primitive could itself be composed of other primitives, with the recursion extending to the pixel level. This concept is an analogy to the concepts of fractal analysis that have become popular recently. Burt et al. have proposed hierarchical texture segmentation schemes based on a pyramidal structure which was discussed in the last chapter [96]. Spann and Wilson [97,98] have used the quad-tree as a structure on which to segment textures. Their algorithm integrated structural and statistical information which drove the segmentation process down the tree.

4.2. Two Recent Approaches to Texture Synthesis

Two very recent approaches to texture discrimination and synthesis will be discussed in more detail. Elements of both of these schemes have been used to synthesise region data using the test sequences described throughout this document.

Volet [99] has used a concept of textures similar to that of Zucker; he considered textures to be formed from a quasi-periodically repeated primitive over the whole region. His synthesis method involved tessellating the texture into a number of small squares. Within each square autocorrelation, median filtering and peak retention are computed to provide the identification of a primitive. Ensembles of the primitives from each small square are used to generate the ideal primitive, and affine transforms [100,101] are derived to supply the local dilation and rotation needed to synthesis the texture over the dimensions of the original in a smoothly varying manner. The synthesised results are very close to the originals; however, the data used are square regions, with edge dimension much larger than the period of the texture. In practice, such as in the data used in this study, the regions are rarely square and the texture primitives may be large compared to the region.

Franke [102-104] uses an approach similar to that of Pratt and Faugeras in order to synthesise non-rectangular textured regions. He takes the power spectrum from an arbitrary textured region, and iteratively deconvolves the texture from the window envelope. His algorithm works iteratively, extracting the largest spectral pair from the DFT of the autocorrelation over the region, reconstructing the texture with a filter controlled by the extracted spectral lines, applying the reverse transform, subtracting and then repeating. Eventually a texture is synthesised, represented by a fixed number of spectral lines in the spatial-frequency, which provides an acceptable approximation to the original over the region. His results, like those of Volet, are appealing, but the computation required to generate them is expensive, often involving tens of iterations and forward-reverse transform sequences.

4.3. Texture Synthesis for Colour Data Compression

From the last chapter, the regions which required the most transmitted bits were ones in which there was high activity. This activity measure correlates well with texture that appears to be constructed from a quasi-periodic pattern, such as the shirt and the curtain in the "Trevor White" sequence. The goal of texture synthesis within this part of the study is essentially one of finding an alternative region description in situations where the compression of the texture achieved by the quad-tree was considered insufficient. Because of Volet's success in synthesising "regular" textures similar to those discussed above, the use of autocorrelations was investigated initially. It should be noted that both Volet and

Franke investigated achromatic textures only. In this study, the use of colour in the texture synthesis was restricted to the methods used in the last chapter; a single pair of co-ordinates were used to wash the synthesised achromatic textures. This method produced acceptable colour reproduction of the regions selected for synthesis. Colour textures will be discussed further in the concluding paragraphs of this chapter.

4.4. The Autocorrelation Function as an Estimator

There are two autocorrelation estimators found in the literature, known as *biased* and *unbiased* estimators. The biased estimator is defined as

$$c_b(k,l) \equiv \frac{\sum_{i=0}^{K-|k|-1} \sum_{j=0}^{L-|l|-1} x(i,j)\cdot x(k+i,l+j)}{K\cdot L} \qquad (4.1)$$

and the unbiased estimator is defined as

$$c_{ub}(k,l) \equiv \frac{\sum_{i=0}^{K-|k|-1} \sum_{j=0}^{L-|l|-1} x(i,j)\cdot x(k+i,l+j)}{(K-|k|)\cdot(L-|l|)}\ , \qquad (4.2)$$

where ***K*** and ***L*** are the x- and y-dimensions of the (rectangular) area of interest. Hence for a square area of side *N*, there are $N^2\cdot\left(\frac{N+1}{2}\right)^2$ multiplications and additions required to get the autocorrelation, biased or unbiased, i.e. $O(N^4)$ calculations. The unbiased estimator, unlike the biased estimator, makes an allowance for the smaller number of coefficients in the summations as k and l approach the dimensions K and L respectively. The two are related by

$$c_{ub}(k,l) = \frac{K \cdot L}{(K-|k|)\cdot(L-|l|)} \cdot c_b(k,l) \ , \ k<K \ , \ l<L \ . \tag{4.3}$$

There is support for both these estimators in the literature (see, for example, Volet). This study will investigate the use of both in texture synthesis.

4.5. Autocorrelation Estimates from Image Regions

Volet, and Deguchi and Morishta [105] perform an autocorrelation over the entire data (or a large part of it) and use that signal as the basis for further processing. This method is quite practicable when the texture field is rectangular and the dimensions of the texture primitive are much smaller than the edge lengths. Unfortunately, in this study, the regions are the result of a prior segmentation, and these requirements cannot be met. Indeed, the regions are often not even convex polygons, being highly convoluted (as the perimeter follows a contour identified by the segmentation algorithm). The two assumptions which had to be made in order to perform autocorrelations on the data were:

1. It is possible to extract a rectangular area from within the region which is large enough to cover a multiple of the primitives in any direction from the centre of the selected rectangle

and

2. the primitive that is extracted from this rectangle is a reasonable candidate for semi-regular placement throughout the region, i.e. the autocorrelation calculated from this rectangle is a good approximation to that of the signal throughout the region.

Once the candidate rectangle was found for the region, and before the autocorrelation was calculated, it was found to be necessary to contrast enhance the signal in order to improve the peak response. This process involved the following simple steps:

1. Identify the maximum, minimum and average grey-level values in the candidate rectangle.

2. Select the smaller of $\frac{255-max}{max-ave}$ and $\frac{min}{ave-min}$ as an enhancement factor, F.

3. Add $F \cdot (pixel-ave)$ to each pixel.

The assumptions used in this enhancement are that $ave \approx \frac{max+min}{2}$ and $ave \approx 128$, i.e. that the data were reasonably evenly distributed about the middle of the grey-scale. This enhancement led to a much improved autocorrelation signal. The textures that Volet investigated in general produced so many peaks in the autocorrelation that he performed median filtering in order to preserve only the most pronounced; there were typically more than thirty peaks around the origin. The textures investigated in this study did not lead to this sort of signal richness. Indeed, there were usually too few peaks in the 2-d representative autocorrelation to enable extraction of a primitive to proceed. Stanger [106] suggested that 1-d autocorrelations have a rich signal, and implementation of a series of these on the textures being investigated suggested that an alternative to Volet's general algorithm could be used to detect the primitive and determine the placement rules.

4.6. Multiple Autocorrelations for Texture Synthesis

The use of 1-d autocorrelations along rows and columns within the candidate rectangle produced favourable signals with demonstrable peaks. By using a histogram of these signals over the rectangle and selecting the largest peak along the x- and y-axes, it was found that there was good agreement in many instances between the perceived texture primitive and the rectangle extracted from the histogram analysis. The use of both biased and unbiased estimators in these calculations demonstrated than in most instances the unbiased estimator performed far better than the biased estimator. At long correlation lags, however, as

$(K-|k|)\cdot(L-|l|)\rightarrow 1$, the unbiased estimator could often produce spurious signals. Note that, from a computational standpoint, the use of 1-d autocorrelations rather than 2-d reduces the number of arithmetic operations to $2N^2\left(\frac{N+1}{2}\right)$, or $O(N^3)$.

4.7. Texture Primitive and Placement Rules

Having extracted the peaks from his 2-d autocorrelations, Volet selected a number of these to determine basis vectors which describe the local periodicity of the texture primitive. The rules of regular tessellations of polygons (see Fejes-Toth [107], Lord and Wilson [108]), those deformations which leave a shape invariant, were then used to identify the shape of the primitive and the rules for placement. In order to identify placement rules for the primitive extracted from the histogram analysis used in this study, the autocorrelations of individual rows and columns had to be compared. Those in agreement with the ensemble from the histogram were considered to be the best choice. By selecting two rows and two columns, ideally placed as far apart as possible, and by cross-correlating their signals, the basis vectors could be extracted and hence the placement rules calculated. Figure 4.1 illustrates the entire process described above. Note that regular tessellations are not considered here because the only regular tessellation possible for a rectangle results in orthogonal vectors. By selecting basis vectors, two of them, which are not themselves orthogonal, the rectangle is deformed into a parallelogram. Figures 4.2 and 4.3 then illustrate the placement rules associated with this algorithm and the typical relationship between any pixel in the region and the extracted primitive. It is apparent that there are eight parameters extracted from this algorithm, as well as the primitive:

> the origin (x, y) and dimensions (width, height) of the rectangle containing the primitive

and

> the relative co-ordinates (x, y) of the two vectors which determine the placement rules.

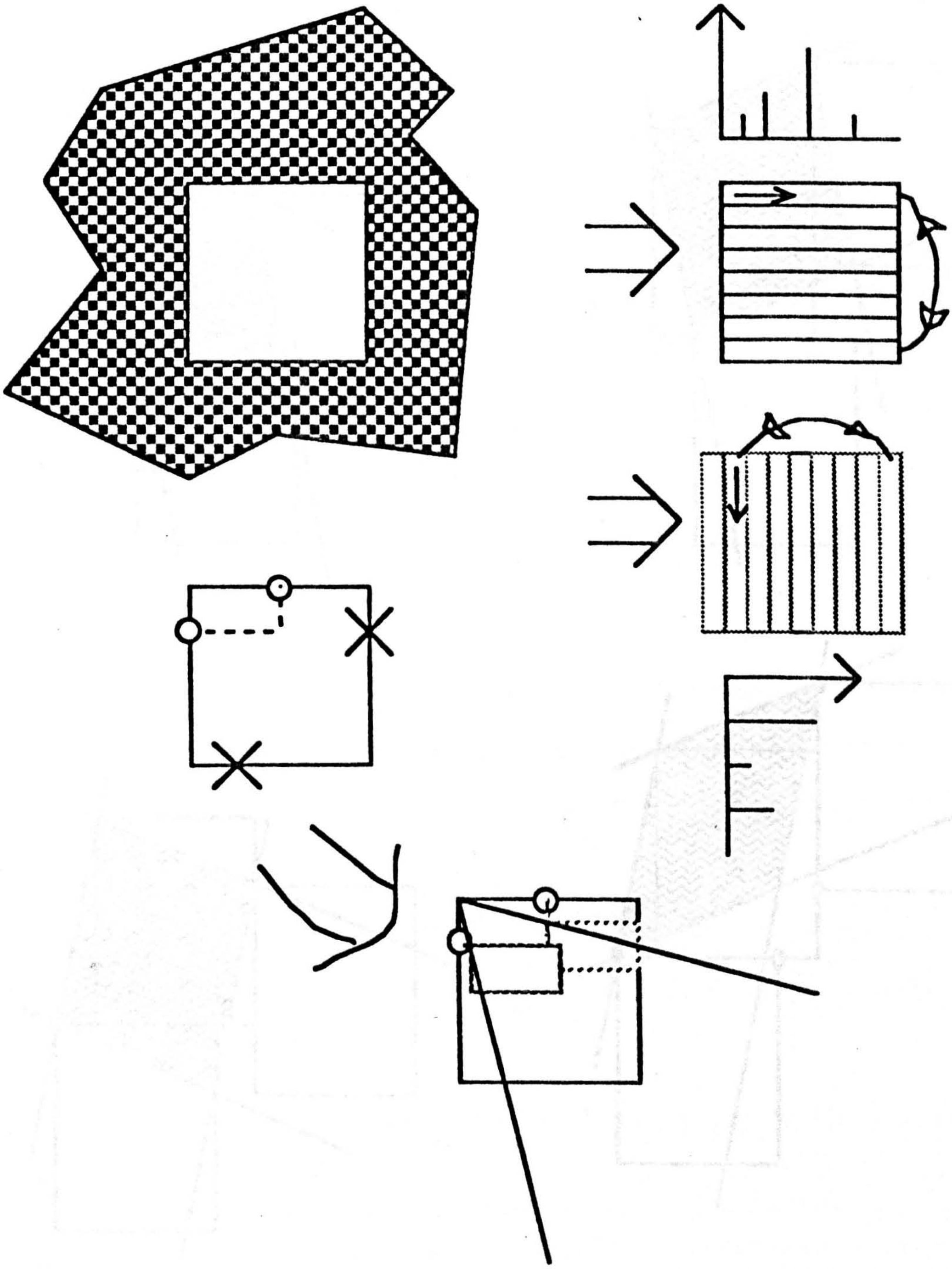


Figure 4.1. Diagrams of the various stages in the texture identification and extraction of placement rules for a region in the image.

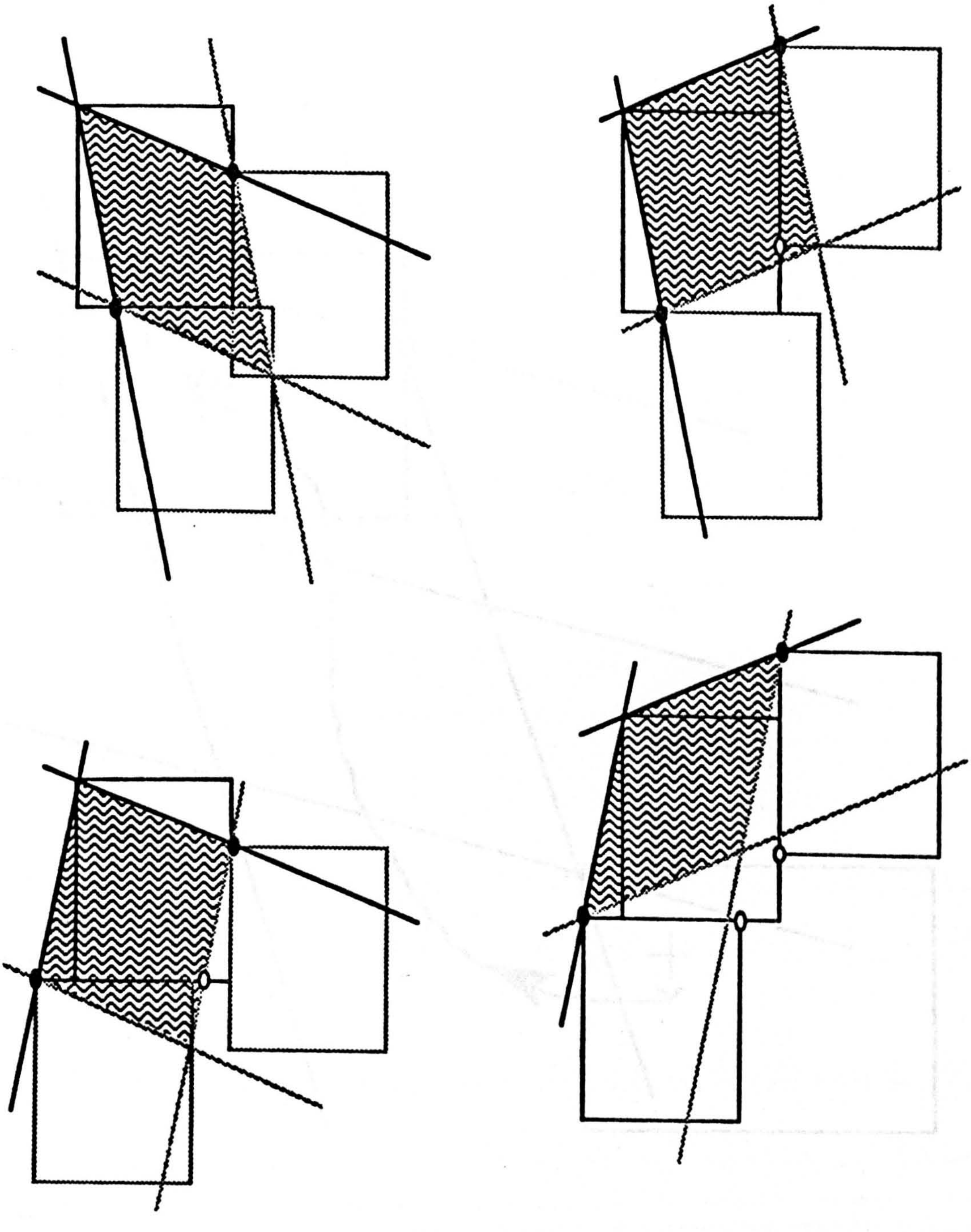


Figure 4.2. Examples of the placement rules according to repetition vectors extracted from texture analysis

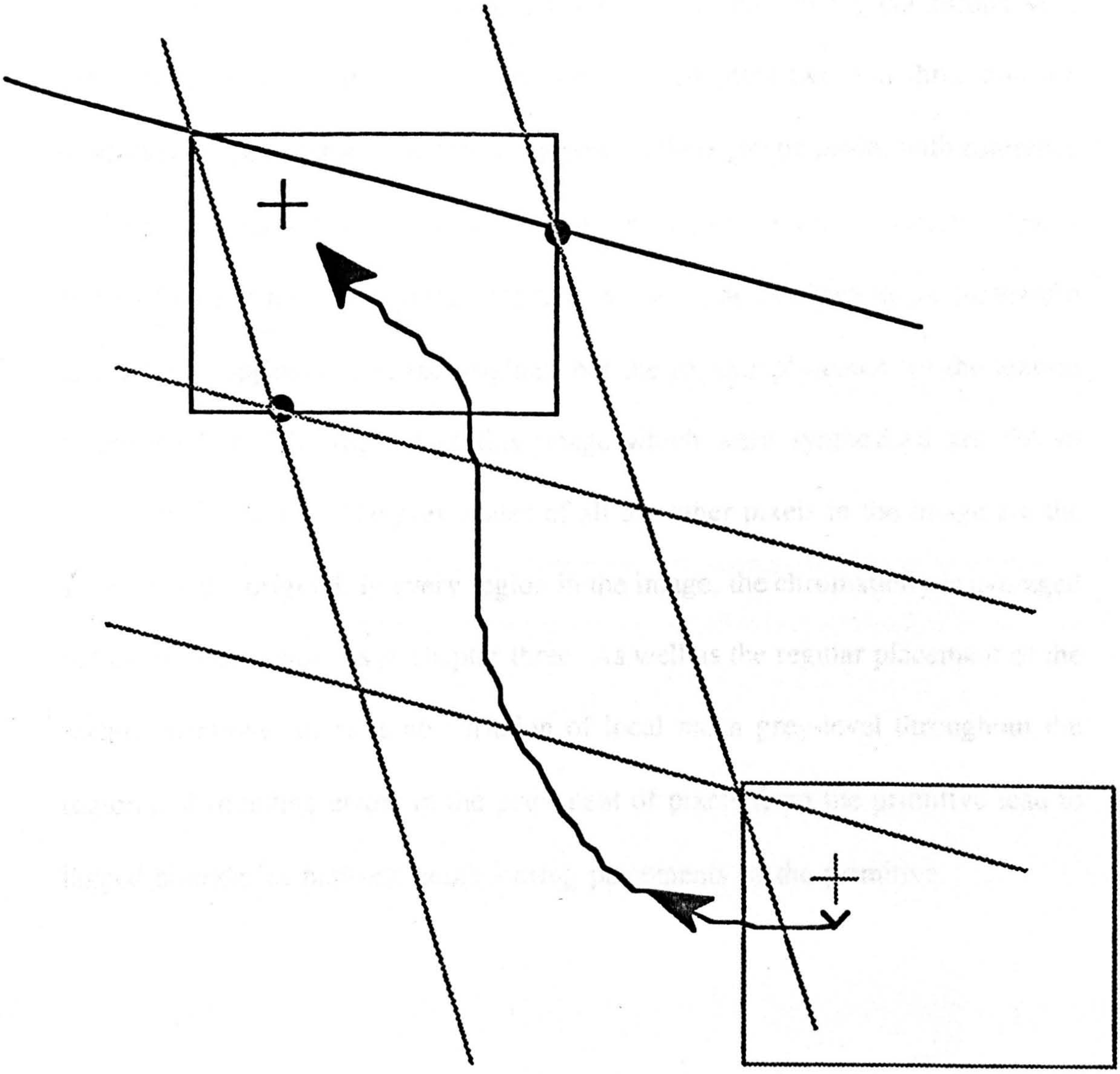


Figure 4.3. Relationship of pixel in synthesised region to pixel found in the texture primitive rectangle.

4.8. Test Results

This algorithm was implemented on suitable regions from the test sequences used throughout this paper. The experimental viewing conditions were the same as those used for the experiments in chapters two and three and are described in appendix A. The following observations can be made, with reference to figures 4.4 and 4.5 which show real and synthesised textures in various regions from a frame of the test sequence "Trevor White". The resulting synthetic texture is similar in appearance to the original, but the regular placement of the texture is unappealing. The regions of this image which were synthesised are shown shaded in figure 4.6. The grey-scales of all the other pixels in the image are the same as in the original. In every region in the image, the chromaticity is averaged before reconstruction, as in chapter three. As well as the regular placement of the texture primitive, there is no variation of local mean grey-level throughout the region and rounding errors in the placement of pixels from the primitive lead to jagged boundaries between neighbouring placements of the primitive.

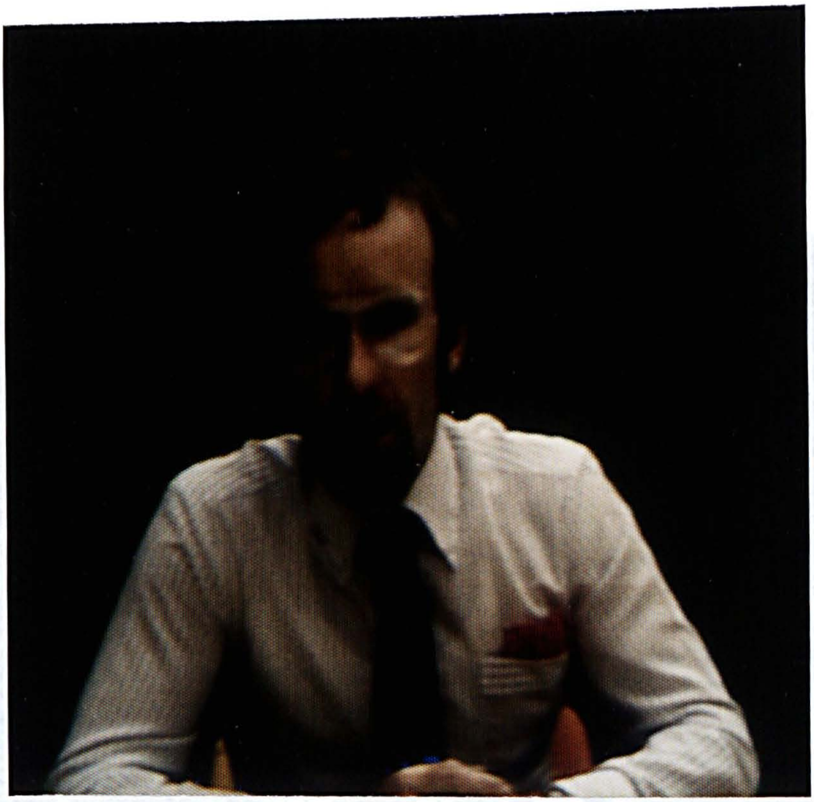


Figure 4.4. Original frame from test sequence "Trevor White".

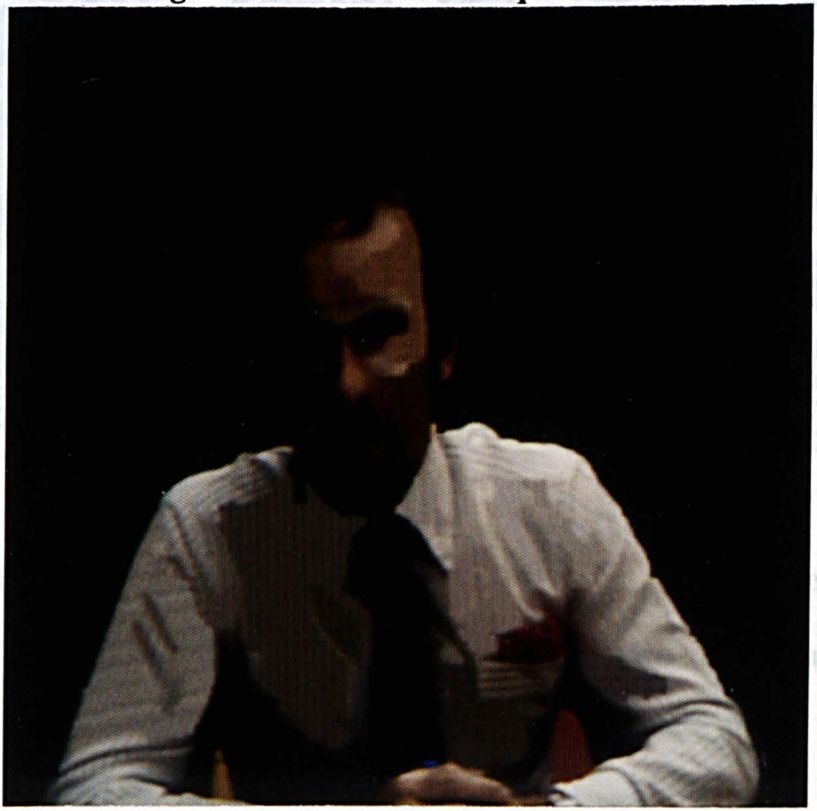


Figure 4.5. Synthesised textures using input from figure 4.4

and the synthesis scheme described above.

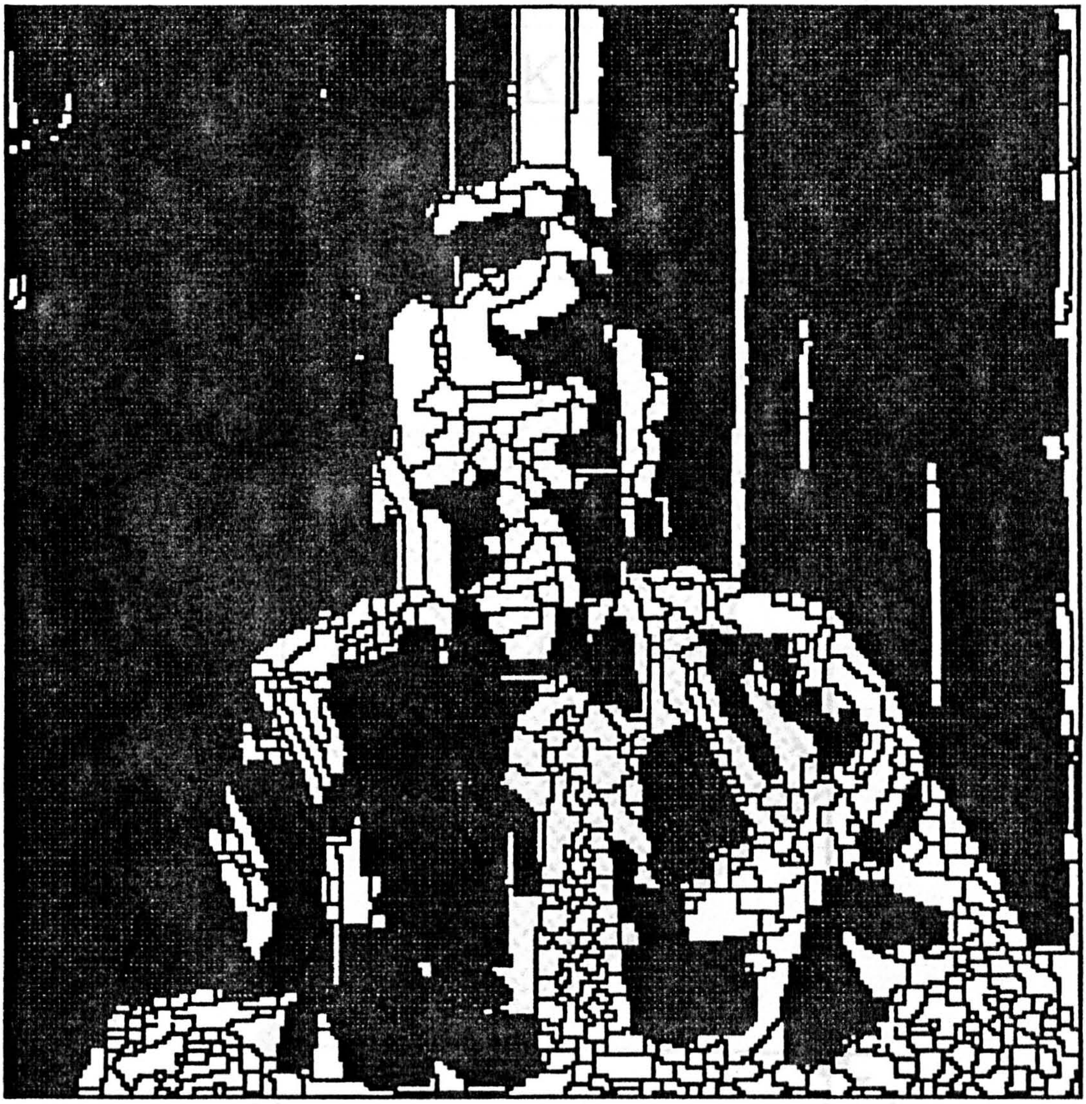
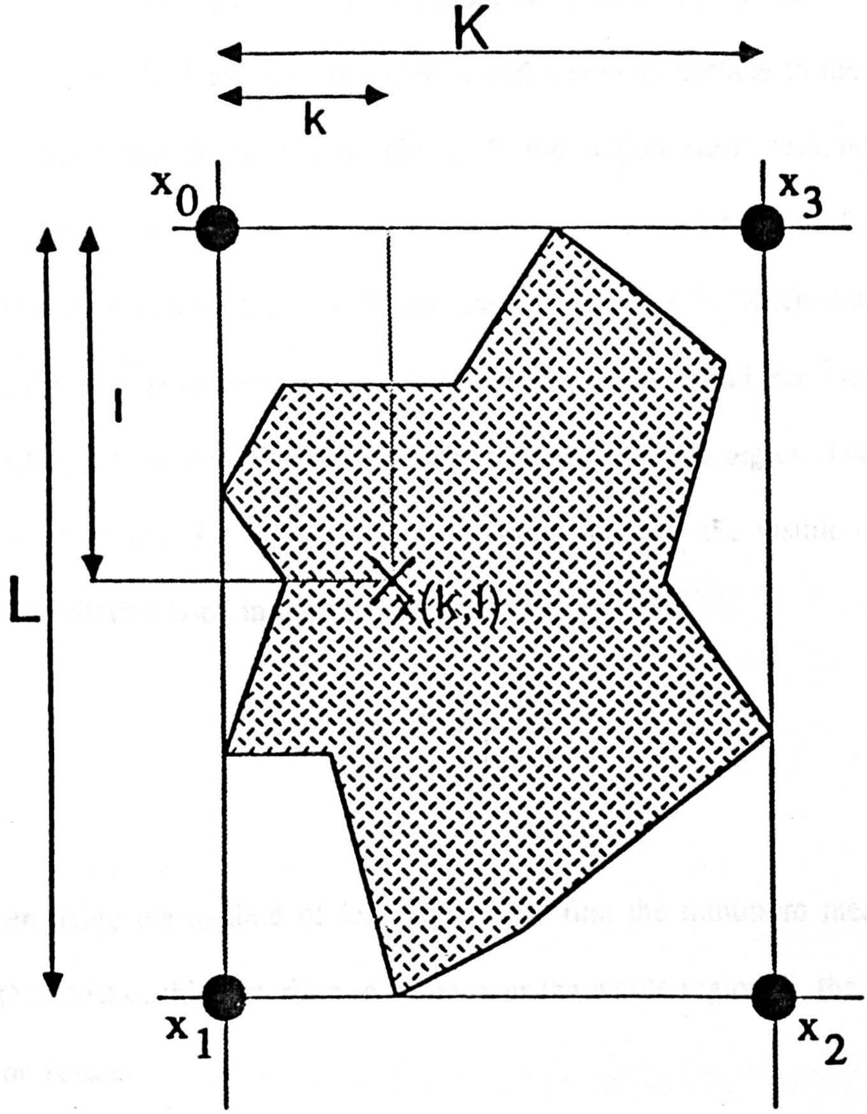


Figure 4.6. Segmentation map of original frame used for region texture synthesis. Those regions in which the texture was synthesised are shaded.

$$\boxed{\hat{x}(k,l) = \sum_{k=0}^{3} d_k \cdot x_k}$$



$$\boxed{\hat{x}(k,l) = \frac{(K-k)(L-l).x_0+k(L-l).x_1+kl.x_2+(K-k)l.x_3}{K.L}}$$

Figure 4.7. Illustration of bilinear interpolation algorithm.

Used by synthesis algorithm to find best-fit corner pixels.

4.9. Bilinear Interpolation for Improved Texture Synthesis

The solution to the problem of fixed mean intensity in the placement of the primitive is to perform some interpolation process across the region, for instance bilinear interpolation, in order to add a smooth surface to the repeated "perturbation" that forms the primitive. If the region were rectangular, the solution would be straightforward. Once again, the region shape makes the algorithm more complicated. With reference to figure 4.7, which includes the general formula for bilinear interpolation, the solution is to find those four corner pixel values which lead to the least error between the true region data and the synthesised texture. By choosing as a distortion measure the visible difference threshold criterion used in the last chapter,

$$D(x,\hat{x}) \equiv \left| \frac{x-\hat{x}}{x} \right| \tag{4.4}$$

and then using the method of least-squares to find the minimum mean square error (MMSE) of this distortion measure over the whole region ***R***, the following equation results:

$$\sum_{i,j\epsilon R} \frac{d_k(i,j)}{x(i,j)} = \sum_{i,j\epsilon R} \frac{d_k(i,j)\cdot\hat{x}(i,j)}{x^2(i,j)}, \quad k{=}0,1,2,3 \tag{4.5}$$

where ***k*** is the index for the 1-d array of the four corner pixels, and $d_k()$ is the distance coefficient for the estimated pixel value associated with the appropriate corner pixel (the expectation is simply a division of both sides by the number of pixels in the region ***R***). The left-hand side of (4.5) is a 1x4 vector of summations, and the right-hand side is the product of a 4x4 and a 1x4 matrix, which will be the solution for the corner pixels, X_k, (the summations apply to every pixel ***x(i,j)*** over ***R***):

$$\begin{bmatrix} \sum \frac{d_0}{x} \\ \sum \frac{d_1}{x} \\ \sum \frac{d_2}{x} \\ \sum \frac{d_3}{x} \end{bmatrix} = \begin{bmatrix} \sum \frac{d_0^2}{x^2} & \sum \frac{d_0 d_1}{x^2} & \sum \frac{d_0 d_2}{x^2} & \sum \frac{d_0 d_3}{x^2} \\ \sum \frac{d_0 d_1}{x^2} & \sum \frac{d_1^2}{x^2} & \sum \frac{d_1 d_2}{x^2} & \sum \frac{d_1 d_3}{x^2} \\ \sum \frac{d_0 d_2}{x^2} & \sum \frac{d_1 d_2}{x^2} & \sum \frac{d_2^2}{x^2} & \sum \frac{d_2 d_3}{x^2} \\ \sum \frac{d_0 d_3}{x^2} & \sum \frac{d_1 d_3}{x^2} & \sum \frac{d_2 d_3}{x^2} & \sum \frac{d_3^2}{x^2} \end{bmatrix} . \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix} . \tag{4.6}$$

Inversion of the 4x4 matrix leads to the desired solution.

Applying the bilinear interpolation alone to the test images produces a more pleasing reproduction of the texture, as can be seen by comparing figure 4.8 with figures 4.4. and 4.5, at the expense of approximately four extra bytes of description. It should be noted that this solution is in fact not the preferred one; the denominator in equation (4.4) is normally the interpolated value, rather than the actual value. The use of the actual value as the denominator was required to make the minimisation tractable in matrix form. The regions which have been synthesised are also those which are shaded in figure 4.6.

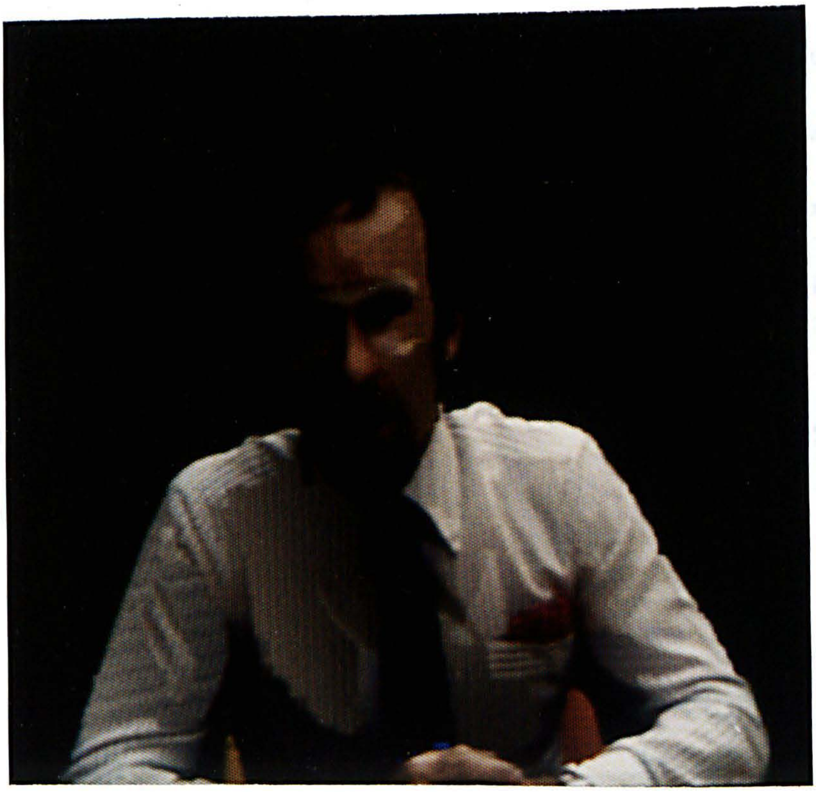


Figure 4.8. Synthesised textures with bilinear interpolation in each region to provide variation of mean intensity level.

The solution to the placement rule, the semi-regular rather than regular placement of the primitive in order to more accurately represent the texture data, is considerably more complex. The technique used by Volet was to derive an 'ideal' texture from the ensemble of candidate primitives, and apply affine transforms from the ideal to the extracted primitives to calculate, for every pixel in the region, the appropriate pixel to use from the ideal primitive and hence generate a smooth variation in the synthesised texture throughout. Because in this study there is no ensemble of candidate primitives this technique cannot be used.

The alternative was to move the primitive around the selected point of placement and try to find the best agreement. Unfortunately this technique still does not allow for rotation and scaling effects. Rotation effects can be incorporated to some extent by simply testing rotated versions of the primitive to find the best local cross-correlation between the primitive and the original, but this is time-consuming. More favourable results using this technique depend on developing suitable rotation and scaling algorithms.

4.10. Cost Comparison of Region Compression Schemes

Remember that the goal of texture synthesis in this study was a greater compression of high activity regions than that obtained from the region-coding algorithm developed in the last chapter, with a similar reconstruction fidelity. The cost, before entropy coding, of synthesising the regions described above will be sum of the following:

1. The origin of the candidate rectangle, with respect to the origin of the region;
2. The dimensions of the candidate rectangle;
3. The two vectors that specify the placement rules;
4. The best-fit corner pixels $\mathbf{X_0}$-$\mathbf{X_3}$;
5. The mean value of the pixels in the candidate rectangle; and

6. The difference between each pixel in the candidate rectangle and the mean of the rectangle.

The first five items listed above can be adequately expressed using 8- to 10-bit quantities, and the pixel differences can be expressed using 4- or 5-bit quantisers, as were used in the last chapter. It is important, however, that the placement rules are coded exactly, otherwise spurious placement vectors may result. Consider figure 4.9, which shows two of the synthesised regions shaded lighter than the rest. These may be labelled **Left of Trevor** and **Right of Trevor.** For each of these regions, the dimensions of the candidate rectangle, and the resulting transmission cost, were significantly smaller than that generated by the quad-tree coder developed in the last chapter. Table 4.1 gives the approximate costs associated with each region. For the quad-tree coder, the numbers given reflect a contrast threshold value of 0.10, which was used previously (see table 3.1 and figure 3.27). The numbers for the synthesis algorithm are raw output; the use of entropy coding could be expected to reduce them by a further ten to thirty percent. For the region **Right of Trevor**, the cost is less than a fifth of that associated with the quad-tree coder; for the region **Left of Trevor**, this ratio is less than 2 percent.

Unfortunately, the compression rate that is achievable with this algorithm is directly dependent on the size of the candidate rectangle which is extracted.

Most of the regions which were synthesised required a much larger number of bits in relation to their size. In general, it seems that the more vivid the texture, the higher the compression that can be achieved. It is unfortunate then that so few of the regions found in the segmented test sequences used meet the criteria needed to apply this synthesis algorithm.

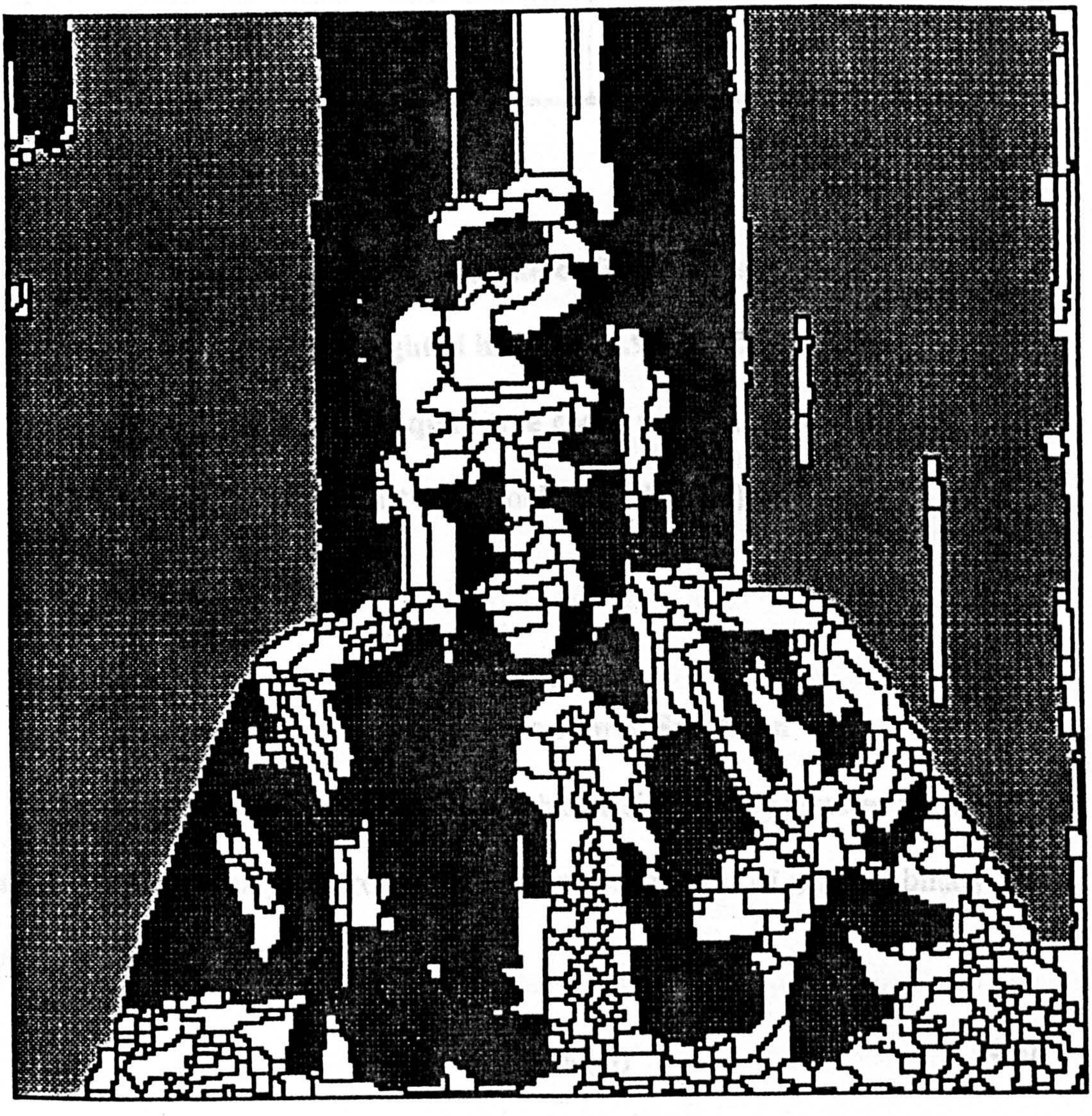


Figure 4.9. Segmented image map highlighting two regions from the "Trevor White" frame used for texture synthesis.

Region	Texture Primitive X-Dim and Y-Dim Pixels	Approx Cost from Synthesis	Q-Coder Cost from Quad-Tree
Left of Trevor	17 x 1	200 bits	15,687 bits
Right of Trevor	15 x 29	2,290 bits	11,626 bits

Table 4.1. Comparisons of approximate coding cost for texture synthesis of the two regions highlighted in figure 4.9 and for the entropy-coded output from the quad-tree coder used in chapter three. (contrast threshold value is 0.10)

4.11. Texture Synthesis from the Power Spectrum

Franke proposes an algorithm which conceives of any textured region *r(i,j)* as the product a complete, rectangular image or texture, *t(i,j)* and a binary window function, *w(i,j)*,

$$r(i,j) = t(i,j)\cdot w(i,j) \ . \qquad (4.5)$$

In the spatial frequency domain, the power spectrum *R(u,v)* is therefore the convolution of the spectrum *T(u,v)* with the spectrum of the window *W(u,v)*,

$$R(u,v) = T(u,v) * W(u,v) \; . \qquad (3.6)$$

His algorithm iteratively computes the spectrum of the "current" windowed, rectangular texture, extracts the largest spectral pair which has not yet been considered; these are added to the new estimate of the deconvolved texture. The result is transformed back to the spatial-domain and so on, until a reasonable texture synthesis is achieved with a limited number of spectral pairs.

The basis to Franke's algorithm was investigated as part of a possible solution to the texture synthesis problem. By simply retaining a few (ten or less) of the peaks found in the DFT of the 2-D autocorrelation, which is also the power spectrum, from selected squares of texture in the "Trevor White" image (curtains, shirt), the reconstructed texture was fairly appealing, even when the peaks did not lie on the orthogonal axes, i.e. when the spatial "direction" of the key textural feature or features was non-orthogonal to the vertical and horizontal axes. It was felt, however, that an iterative solution, which is required to deconvolve the effect of the window, was liable to be too time-consuming for teleconferencing applications. In fact, Franke has proposed this technique for archiving purposes rather than real-time applications.

4.12. Summary

To summarise, the use of autocorrelations, histograms, texture primitives, placement vectors and interpolation techniques can in some instances lead to the successful synthesis of vivid textures such as were found in certain regions of segmented images from the test sequences. Unfortunately, however, the complex polygonal region shape complicates the identification of the primitive and the computation of the placement rules. In addition, the lack of provision for scaling and rotation components in the placement rules leads only to a regular placement of the primitive, rather than the quasi-regular or aperiodic placement strategy proposed elsewhere, hence the results are somewhat less appealing. Low pass filtering the synthesised texture to reduce the impact of such artifacts may be a solution but it is felt that this runs contrary to the goal, and the tendency to smear true edges is unlikely to improve the perceived quality.

The chromatic components were synthesised as per the scheme discussed in the previous chapter. The fact that a single pair of chromaticity co-ordinates could be used to reconstruct accurately (from a subjective standpoint) the colour of the texture could be due to the fact that the achromatic textures synthesised were vivid, leading to a loss of sensitivity to chromatic errors by the visual system, especially if there is strong evidence of an achromatic distortion.

Chapter 5. Conclusions

Throughout this thesis, the important role of human perception in color image processing has been stressed. It is apparent that a reliable, automatic method with which to specify the quality of a reconstructed colour image, when it is compared with the original, has still to be found. It may not even be possible to specify accurately the quantity of colour image distortion in a way that agrees, in an exactly linear fashion, with our perception of it. Schemes have been presented in this thesis to quantise, encode and synthesise the content of typical colour images encountered in the somewhat restricted world of videoconferencing. I feel that the algorithms and results presented in chapters two, three and four have at least achieved the goal of an improvement on other algorithms reported in the literature, from the standpoint of perceived image quality versus the quantisation cost or compression ratios achieved. Of course, there are a number of possible improvements which can be made to these algorithms.

In the next seven sections, I will first summarise the contents of, and algorithms developed in, each of chapters two, three and four, then I will briefly revisit particular points which were discussed in each of these chapters. I will mention the failings of the appropriate algorithms and the improvements that hopefully can be made to these schemes. Some improvements may be implemented at little cost; others present considerable challenges, if the basic framework of the algorithm is to be preserved. Such future work that is discussed for this project will be included within each section. A final section will briefly discuss the future trends of video communications technology up to the end of this century.

5.1. Summary of Chapter 2 - Quantisation of Colour Image Data

In chapter two the aim was to develop an algorithm which would allow the quantisation of any digitised colour image in such a way that any distortions introduced in the image were such that the perceived distortion was minimised. MacAdam's geodesic was selected as a chromaticity space in which perceived colour difference is related to Euclidean distance on the geodesic. A clustering algorithm, the k-means test, was used to generate a relaxed solution to the choice of quantised outputs for luminance and chromaticity. Three variations on the basic quantiser design were compared for reconstructed quality. The Mean Square and Mean Absolute Errors (MSE and MAE) were calculated to relate the chromaticity distortions to subjective evaluation of the reconstructions. The final quantisation was 4.75 bits per pixel, or a compression ratio of approximately 6:1. Original contributions in this chapter include the use of clustering, the design of the LUT and the distortion analysis.

5.2. Summary of Chapter 3 - Two-Component Colour Image Coding

In chapter three I developed an algorithm to improve the compression and quality of those areas in digitised colour images identified as regions, areas of almost constant or slowly changing intensity, which are in turn separated from other regions by contours or edges, those points where the image properties change quickly. Rather than using a polynomial model for the data in the region, the algorithm developed from the use of a partially filled quad-tree, constructed from the luminance data in the region. A simple contrast threshold criterion for node transmission was developed, using the Weber fraction. The chromaticity data for the region was simply an average over the region, leading to a colour washing process at the decoder, wherein the reconstructed luminance values were modulated by the chromaticity to generate the resulting pixel colour. Arithmetic coding techniques were also employed to further compress the data transmitted as the description of each region. A number of segmented colour images were used to test the region-coding algorithm at various compression rates. The highest compression achieved from the test images was over 48:1, at a rate of 0.494 bits per pixel. The original contributions in this chapter include the development of the quad-tree based region coder and the contrast threshold node transmission criterion.

5.3. Summary of Chapter 4 - Textures: Analysis and Synthesis

In chapter four, a structural and statistical description was sought of the textures found in certain regions of the segmented images processed in chapter three, in order to provide further compression of the region data by a process of identification, extraction and synthesis of the texture in the region. Autocorrelation functions, the grey-scale histogram, contrast thresholding and bilinear interpolation were all used to develop a texture synthesis algorithm which could extract and repeat a texture primitive over a region of a segmented colour image. Comparisons were made between the costs of this scheme and the one developed in chapter three. The use of the power spectrum was also discussed as an alternative signal space in which to develop texture synthesis. For certain image regions, compressions of more than fifty times that available with the quad-tree region coder were achieved. The original contributions in this chapter include the development of the identification scheme for the candidate rectangle, the scheme to identify and tile the extracted region primitive and the algorithm to smooth the luminance variation by bilinear interpolation.

5.4. Distortion Measure - The MSE and Human Perception

It was reported in the introduction that probably the most common distortion measure used to compare processed images is the MSE. This measure is used because of its mathematical tractability, despite the fact that it is unlikely that the visual system is capable of performing the computations involved in the MSE, and that a single measure over an entire image does not in general agree with perceived errors, which are local phenomena. Faugeras [29] argues that perhaps, of the three colour primaries, the one with the largest error should be used, or simply the summed absolute differences. Referring to tables 2.1 and 2.2, which compare the Mean Square- and Mean Absolute errors for the three quantisation methods developed in chapter two, there is a good agreement between perceived reconstruction quality and these measures, but it is hard to state unequivocally that the relationship is strictly linear. At best it could be described as monotonic. Limb certainly finds this trend with the RMSE [112].

I reported in chapter two that the use of the MacAdam geodesic as a chromaticity chart permitted the use of the MSE in a perceptually meaningful way. In fact, to calculate the Euclidean distance, we require the square root of this measure. As Watson [30] points out, the human visual system is unlikely to be capable of computing square roots, hence there must be a suspicion that the use of the MacAdam chart for quantitative distortion analyses is inappropriate. A suggestion that of course comes to mind is to take the reconstructed image in a colour space such as UVW or XYZ and use perhaps the MAE as a distortion measure.

There is another factor involved in this discussion - that of aesthetics. Certain intended (or unintended) distortions present in reconstructed images may be perceived as being more visually appealing than others. This is a complex phenomenon which inevitably occurs at the man-machine interface. Because of this facet to the problem, any future work on automated distortion measures must identify how much influence our sense of aesthetism has on making objective measurements. The influence of this factor is likely to increase with the severity of the introduced distortions.

5.5. Sensitivity to Luminance - Logarithmic Relationships

In chapter two a number of sources were quoted, who suggested that the visual system exhibits a linear sensitivity to a function of the absolute luminance which can be approximated by that of the natural logarithm. Faugeras [29] goes further and assumes that all three of the visual processes exhibit a logarithmic sensitivity to light at an early stage in the visual pathway. Yet the algorithm developed in chapter two used the absolute value of the achromatic data as the source to the clustering process. The reason for this was empirical; there was an observed improvement in the output of the quantiser when log-scale data were replaced by their absolute equivalents. Therefore as the clustering process was applied to the achromatic content of each image, minimisations were being found, in a Euclidean sense, over a histogram of absolute values. It has already been mentioned that the visual system is unlikely to be able to calculate Euclidean distances. Therefore, there is no more basis, from the standpoint of physiological plausibility, for running k-means tests on histograms of logarithms than on histograms of absolute intensities.

By using as the "just noticeable contrast difference" threshold the Weber fraction, which was introduced in chapter three, an improvement to the clustering algorithm may bė achieved. By using this method, the distance of any particular index within the gray-scale histogram will be measured from each centroid by the ratio of their difference to the absolute value of one or the other - normally the denominator is the centroid. This would bring the quantisation algorithm into the same framework as the other algorithms implemented throughout this thesis. Such a modification would involve a minor alteration to the achromatic clustering algorithm and hence could be easily implemented. Further improvements to the threshold model should be incorporation of the asymptotic behaviour exhibited at very high and very low background intensity values. The effect of the experimental equipment, such as monitor gamma - which was not investigated in this research - should also be considered; this may provide an explanation for the achromatic clustering results in chapter two and also improve the contrast threshold model. This model will be discussed further in the next section.

5.6. The "Just Noticeable Contrast Difference" Threshold

In chapter three, the Weber fraction was used as a simple threshold operator to control quad-tree encoding of region data. The results of using this region coding algorithm were a definite improvement over those from existing region coding models, but it was observed that the Weber fraction is a poor measure of the true contrast in a "real" image; indeed Peli and Goldstein [108] point out that it is hard to define the contrast in complex images, because contrast is a local phenomenon. In chapter four it was shown that computing the spatial-frequency content of data contained in a convoluted, polygonal region is very difficult. With this information, it can be expected that the inclusion of a frequency-dependant function, such as that shown in figure 3.5, in the threshold operator, is not really tractable. This is not a fatal drawback, however, if it is borne in mind that the thresholding measurements are made on the relationships between nodes in the quad-tree. What is required instead is the development of some measure of region activity, using the data obtained from the nodes on the quad-tree, which may then be used to modify the behaviour of the thresholding process. It is easy, for instance, to find the "energy", or variance, of any node given the values of its children.

In the absence of a more suitable modification to the threshold operator, then, the incorporation of high- and low-intensity asymptotes, together with an applied increment with decreasing level on the tree (as the coder descends to the leaves), are simple modifications that can be implemented quickly and may be expected to provide some improvement in compression ratio and/or reconstruction quality. Increasing the complexity of this operator should involve a note of caution, however. Since the segmentation process, which is computationally intensive, has already created the regions, this data should be almost free of discontinuity or rapid change; the region coder should not be attempting to do the job of the segmentation algorithm. There will always competition, and hence a tradeoff of this nature, involved in the two-component coder.

5.7. Texture Synthesis - Further Developments

The results obtained from the texture synthesis algorithm developed in chapter four have, arguably, a limited impact as an improvement to the region coding scheme developed in chapter three. Only those regions in which it is possible to identify a simple, regularly repeated, primitive can be used on this algorithm, and only in the "Trevor White" sequence do such textures exist. This sequence is recognised as being at the high end of the scale of "coding difficulty". The results shown in table 3.1. reflect this statement.

It is unfortunate that the regions generated as a result of segmentation in the spatial domain cannot be easily transformed to yield spatial-frequency domain information. By computing the amplitude spectrum over rectangular areas of the curtain in the "Trevor White" sequence, experiments demonstrated that it was possible to reconstruct the texture accurately with as few as eight of the largest spectral pairs, or about ten per cent of the original signal energy. This scheme of course ignores the boundaries imposed by the segmentation, hence it is not practical within the framework of the two-component model.

Any future work on texture synthesis involving the basic algorithm in chapter four must include the development of semi-periodic placement rules to cope with scaling and rotation effects. I discussed how these effects may be identified in chapter four. The incorporation of this behaviour into the synthesis algorithm will allow a much more life-like synthesis.

5.8. The Role of the Chromatic Component

The aim of this thesis has been to investigate the role of the chromatic component in colour image processing, and in our perception of image content modified by such processing. The results presented in chapters two, three and four imply that we are considerably less sensitive to errors in the chromatic components of the reconstructions than we are to errors in the achromatic component. These events are not independent however; most combinations of achromatic and chromatic distortions will result in a perceived colour distortion which, in turn, is not simply proportional to the superposition of the one on the other (assuming we can represent both as scalars). Even if the two could be measured on a metric perceptual scale, the sum of such distortions over an image is still unlikely to agree with subjective tests in every instance, for the reasons given in section 5.1.

What has been achieved in this project, I feel, is the development of a technique for the compression of the chromatic information in colour images that effectively employs perceptual criteria to minimise observed image distortions. The main direction for the future work in this area should be in establishing the dependencies between introduced achromatic distortions, chromatic distortions and perceived colour changes that result from one or the other. In this way, the achromatic quantisation and quad-tree coding algorithms will be modified to incorporate feedback which modifies the chromatic quantisation and coding algorithms, and vice versa.

5.9. Video Communications in the 1990s

In the 1990s, we can look forward to schemes which provide reasonable quality video communications at transmission rates which are fractions of what has been reported in this thesis [110,111], but such algorithms, which model each image frame as a three-dimensional scene and incorporate concepts of artificial intelligence and advanced computer graphics, must first overcome a series of technical challenges. The increasing availability of ISDN services to commerce and the domestic consumer lends the development and commercial exploitation of low-cost video-communications an appropriate urgency, hence we may expect novel technical improvements to continue to appear. Overcoming the social challenges imposed by this technology is, of course, another story.

References

[1] Young, T., *Lectures on Natural Philosophy*, Vol. 1, p. 440, 1807.

[2] von Helmholtz, *Treatise on Physiological Optics* (The Optical Society of America) Vol. 2, pp. 145,235,240,412, 1924.

[3] MacAdam, D. L., "Color Essays," *J. Opt. Soc. Am.*, Vol. 65, No. 5, pp. 483-492, May 1975.

[4] Judd, D. B., "Fundamental Studies of Color Vision From 1860 to 1960," *Nat. Acad. of Sci.*, Vol. 55, No. 6, pp. 1311-1330, June 1966.

[5] MacAdam, D. L., *Sources of Color Science*, Cambridge, MA, 1970.

[6] Konig, A, "Die Abhangigkeit der Farben- und Helligkeitsgleichen von der absoluten Intensitat," *Sitzber. Akad. Wiss. Berlin*, p. 871, July 20, 1897.

[7] Hering, E., *Zur Lehre vom Lichtsinn*, Berlin, 1878.

[8] Wald, G., "The Receptors for Human Color Vision," *Science*, Vol. 145, No. 3636, pp. 1007-1017, September 4, 1964.

[9] CIE (Commission Internationale de l'Eclairage), *Proc. 8th Session*, pp. 19-29, Cambridge, England, 1931.

[10] Wright, W. D., "A re-determination of the mixture curves of the spectrum," *Trans. Opt. Soc.*, Vol. 31, p. 201, 1929.

[11] Guild, J., "The colorimetric properties of the spectrum," *Phil. Trans. Roy. Soc. London A*, Vol. 230, p. 149, 1931.

[12] Grassman, H. G., "Theory of Compound Colors," *Philosophic Magazine*, Vol. 4, No. 7, pp. 254-264, 1854.

[13] MacAdam, D. L., "On the Geometry of Color Space," *J. Franklin Inst.*, Vol. 238, pp. 195-210, 1944.

[14] MacAdam, D. L., "Geodesic Chromaticity Diagram Based on Variances of Color Matching by 14 Normal Observers," *Appl. Opt.*, Vol. 10, No. 1, pp. 1-7, January 1971.

[15] MacAdam, D. L., "Projective Transformations of ICI Color Specification," *J. Opt. Soc. Am.*, Vol. 27, pp. 294-299, August 1937.

[16] Anonymous, "Technical Note: CIE Colorimetry Committee - Working Program on Color Differences," *J. Opt. Soc. Am.*, Vol. 64, pp. 896-897, June 1974.

[17] Kurz, B., "Optimal Color Quantization for Color Displays," *Proc. IEEE Computer Soc. Conf. on Comp. Vis. and Patt. Rec.*, pp. 217-224, Washington DC, 19-23 June 1983.

[18] Munsell, A. H., *Color Notation*, Munsell Color Company Inc., Baltimore, Maryland, 1946.

[19] Hurvich, L. M., and Jameson, D., "An Opponent-Process Theory of Color Vision," *Psych. Rev.*, Vol. 64, No. 6, pp. 384-404, 1957.

[20] Ohta, Y-I., Kanade, T. and Sakai, T., "Color Information for Region Segmentation," *Comp. Graph. Im. Proc.*, Vol. 13, pp. 222-241, 1980.

[21] Buchsbaum, G. and Gottschalk, A., "Trichromacy, opponent colors coding and optimum colour information transmission in the retina," *Phil. Trans. Roy. Soc. London B*, Vol. 220, pp. 89-113, 1983.

[22] Max, J., "Quantizing for Minimum Distortion ," *IRE Trans. Info. Theory*, Vol. IT-6, No. 1, pp. 7-15, January 1960.

[23] Linde, Y., Buzo, A., and Gray, R. M., "An Algorithm for Vector Quantizer Design," *IEEE Trans. Commun.*, Vol. COM-28, No. 1, pp. 84-95, January 1980.

[24] Stevens, R. J., and Preston, F. H., "Data Ordering and Compression of Multispectral Images Using the Peano Scan," *IEE Int. Conf. on Electron. Im. Proc.*, pp. 194-198, York, England, July 1982.

[25] Lehar, A. F. and Stevens, R. J., "High-Speed Manipulation of the Color Chromaticity of Digital Images," *IEEE Comp. Graph. Appl.*, pp. 34-39, February 1984.

[26] Limb, J. O., Rubinstein, C. B. and Thompson, J. E., "Digital Coding of Color Video Signals - A Review," *IEEE Trans. Commun.*, Vol COM-25, No. 11, pp. 1349-1384, November 1977.

[27] Kunt, M., Ikonomopoulos, A. and Kocher, M., "Second-Generation Image-Coding Techniques," *Proc. IEEE*, Vol. 73, No. 4, pp. 549-574, April 1985.

[28] Schreiber, W. F., Knapp, C. F. and Kay, N. D., "Synthetic highs, an experimental TV bandwidth reduction system," *J. SMPTE*, Vol. 68, pp. 525-537, August 1959.

[29] Faugeras, O. D., "Digital Color Image Processing Within the Framework of a Human Visual Model," *IEEE Trans. Acoustics, Speech and Sig. Proc.*, Vol. ASSP-27, No. 4, pp. 380-393, August 1979.

[30] Watson, A. B., "Efficiency of a model human image code," *J. Opt. Soc. Am. A*, Vol. 4, No. 12, pp. 2401-2417, December 1987.

[31] Stockham, T. G. Jr., "Image Processing in the Context of a Visual Model," *Proc. IEEE*, Vol. 60, No. 7, pp. 828-841, July 1972.

[32] Stenger, L., "Quantization of TV Chrominance Signals Considering the Visibility of Small Color Differences," *IEEE Trans. Commun.*, Vol. COM-25, No. 11, pp. 1393-1406, November 1977.

[33] Heckbert, P., "Color Image Quantization for Frame Buffer Display," *Computer Graphics*, Vol. 16, No. 3, pp. 297-307, July 1982.

[34] Lena, M. and Mitchell, O. R., "Absolute Moment Block Truncation Coding and its Application to Color Images," *IEEE Trans. Commun.*, Vol. COM-32, No. 10, pp. 1148-1157, October 1984.

[35] Wyszecki, G. and Stiles, W. G., *Color Science*, Wiley, New York 1967.

[36] Durret, H. J., *Color and the Computer*, Academic Press, Orlando, Florida 1987.

[37] Fink, D. G. ed, *Color Television Standards: Selected Papers and Records of the National Television Systems Committee*, McGraw-Hill, New York 1955.

[38] MacQueen, J., "Some Methods for Classification and Analysis of Multivariate Observations," *Proc. Fifth Berkeley Symp. on Math., Stat. and Prob.*, Vol. 1, pp. 281-296, 1967.

[39] Pratt, W. H., *Digital Image Processing*, Addison Wesley, New York, 1984.

[40] Hecht, S., "The Visual Discrimination of Intensity and the Weber-Fechner Law," *J. Gen. Physiol.*, Vol. 7, p. 241, 1924.

[41] Cornsweet, T. N., *Visual Perception*, Academic Press, New York 1970.

[42] Michelson, A. A., *Studies in Optics*, University of Chicago Press, Chicago, 1927.

[43] Hunt, "Digital Image Processing," *Proc. IEEE*, Vol. 63, No. 4, pp. 693-708, April 1975.

[44] Mannos, J. L. and Sakrison, D. J., "The Effects of a Visual Fidelity Criterion on the Encoding of Images," *IEEE Trans. Info. Theory*, Vol. IT-20, No. 4, pp. 525-536, July 1974.

[45] Stevens, S. S., "The psychophysics of sensory function," in *Sensory Communication*, (ed. Rosenblith, W. A.), MIT Press, New York 1961.

[46] Pearson, D. E., *Transmission and Display of Pictorial Information*, Pentech Press, 1975.

[47] Wright, W. D., *The Measurement of Colour*, Hilger, London 1969.

[48] Carnt, P. G. and Townsend, G. B., *Colour Television - Volume 2*, Iliffe, London 1969.

[49] McColl, R. W. and Martin, G. R., "Compression of colour image data using histogram analysis and clustering techniques," *IEE Electron. Commun. Eng. J.*, Vol. 1, No. 2, pp. 93-100, March/April 1989.

[50] Young, T. Y. and FU, K-S. (Eds.), *Handbook of pattern recognition and image processing*, Academic Press, London 1986.

[51] Mason, J. R. M., "Texture analysis for the purpose of low bit-rate digital image transmission and storage," *IEE Int. Conf. on Electron. Im. Proc.*, pp. 47-50, York, England, July 1982.

[52] Kocher, M. and Kunt, M., "Image data compression by contour texture modelling," *Applications of Digital Image Processing*, pp. 132-139, 1983.

[53] Silverman, J. and Cooper, D. B., "Bayesian Clustering for Unsupervised Estimation of Surface and Texture Models," *IEEE Trans. Patt. Anal. and Mach. Intell.*, Vol. PAMI-10, No. 4, pp. 482-495, July 1982.

[54] Marr, D. and Hildreth, E., Theory of Edge Detection, *MIT AI Memo* 518, 1979.

[55] Canny, J., "A Computational Approach to Edge Detection," *IEEE Trans. Pat. Rec. and Mach. Intell.*, Vol. PAMI-8, No. 6, pp. 679-698, 1986.

[56] Stanger, V. J. and Symons, A. E., "The Application of Image Analysis Techniques to Low Bit Rate Coding of Color Video-Conferencing Sequences," *GEC Hirst Research Centre Report*, Wembley UK 1987.

[57] Burt, P. J. and Adelson, E. H., "The Laplacian Pyramid as a Compact Image Code," *IEEE Trans. Commun.*, Vol. COM-31, pp. 532-540, April 1983.

[58] Wilson, R. G., "Quad-tree predictive coding: a new class of image data compression algorithms," *Linkoping University Internal Report LiTH-ISY-I-0609*, Linkoping, Sweden 1983.

[59] Tanimoto, S. L. and Pavlidis, "A Hierarchical Data Structure for Picture Processing," *Comp. Graph. and Im. Proc.*, Vol. 4, 1975.

[60] Papoullis, *Probability, Random Variables, and Stochastic Processes*, 2nd. Edition, McGraw-Hill, 1984.

[61] McColl, R. W. and Martin, G. R., "Quad-Tree Modelling of Colour Image Regions," *SPIE Vol. 1001 Vis. Commun. Im. Proc.*, pp. 231-238, Cambridge, MA, November 1988.

[62] McIlwain, K., "Requisite Color Bandwidth for Simultaneous Color-Television Systems," *Proc. IRE*, Vol. 40, pp. 909-921, August 1952.

[63] Clippingdale, S. C. and Wilson, R. G., "Quad-Tree Image Estimation: A New Image Model and its application to Minimum Mean Square Error Image Restoration," *Proc. 5th Scand. Conf. on Im. Anal.*, pp. 699-706, Stockholm, Sweden 1987.

[64] Kelly, D. H., "Visual Contrast Sensitivity," *Optica Acta*, Vol. 24, pp. 107-129, 1977.

[65] Robson, J. G., "Spatial and Temporal Contrast Sensitivity Functions of the Visual System," *J. Opt. Soc. Am.*, Vol. 56, pp. 1141-1142, August 1966.

[66] Huffman, D. A., "A method for the construction of minimum redundancy codes," *Proc. IRE*, Vol. 40, No. 9, pp. 1098-1101, September 1952.

[67] Rissanen, J. J. and Langdon, G. G., "Arithmetic Coding," *IBM J. Res. Develop.*, Vol. 23, No. 2, pp. 149-162, 1979.

[68] Mitchell, J. L. and Pennebaker, W. B., "Software Development of the Q-Coder," *IBM Research Report RC 12660*, IBM Thos. J. Watson Res. Ctr., New York 1987.

[69] Mitchell, J. L. and Pennebaker, W. B., "Probability Estimation for the Q-Coder," *IBM Research Report RC 12659*, IBM Thos. J. Watson Res. Ctr., New York 1987.

[70] Langdon, G. G., "An Introduction to Arithmetic Coding," *IBM J. Res. Develop.*, Vol. 28, No. 2, pp. 135-149, 1984.

[71] Vitter, J. S., "Design and Analysis of Dynamic Huffman Codes," *J. Assoc. Comput. Mach.*, Vol. 34, No. 4, pp. 825-845, October 1987.

[72] Stanger, V. J., "A comparative study of practical image segmentation techniques," *Proc. 5th Scand. Conf. Im. Anal.*, pp. 65-69, Stockholm, Sweden 1987.

[73] Sheldon. K, "A Film of a Different Color," *BYTE*, pp. 164-165, March 1987.

[74] Symons, A.E., GEC Hirst Research Centre, England, Private Communication.

[75] Faugeras, O. D. and Pratt, W. K., "Decorrelation Methods of Texture Feature Extraction," *IEEE Trans. Patt. Anal. and Mach. Intell.*, Vol. PAMI-2, No. 4, pp. 323-332, July 1980.

[76] Davis, L. S., "Image Texture Analysis Techniques - A Survey," in *Digital Image Processing* (eds. Simon, J.C. and Haralick, R. M.), pp. 189-201, 1981.

[77] Julesz, B., "Visual Pattern Discrimination," *IRE Trans. Info. Theory*, Vol. IT-8, pp. 84-92, February 1962.

[78] Julesz, B., "Spatial nonlinearities in the instantaneous perception of textures with identical power," *Phil Trans. Roy. Soc. London B*, Vol. 290, pp. 83-94, 1980.

[79] Julesz, B., "Textons, the elements of texture perception, and their interaction," *Nature*, Vol. 290, pp. 91-97, 12th March 1981.

[80] Voorhees, H. and Poggio, T., "Computing texture boundaries from images," *Nature*, Vol. 333, pp. 364-367, 26th May 1988.

[81] Pratt, W. K., Faugeras, O. D. and Gagalowicz, A., "applications of Stochastic Texture Field Models to Image Processing," *Proc. IEEE*, Vol. 69, No. 5, pp. 542-551, May 1981.

[82] Rosenfeld, A. and Kak, A. C., *Digital Picture Processing (2nd Ed.)*, Academic Press, 1982.

[83] Haralick, R. M., Shanmugan, K. and Dinstein, I., "Textural Features for Image Classification," *IEEE Trans. Sys., Man, and Cybern.*, Vol. SMC-3, No. 6, pp. 610-621, November 1973.

[84] Haralick, R. M. and Shanmugan, K., "Computer classification of reservoir sandstones," *IEEE Trans. Geosci. Electron.*, Vol. GE-11, pp. 171-177, October 1973.

[85] Haralick, R. M., "Statistical and Structural Approaches to Texture," *Proc. IEEE*, Vol. 67, No. 5, pp. 786-803, May 1979.

[86] Modestino, J. W., Fries, R. W. and Vickers, A. L., "Texture Discrimination Based on an Assumed Stochastic Texture Model," *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol. PAMI-3, No. 5, pp. 557-580, September 1981.

[87] Shen, H. C. and Wong, A. K. C., "Generalized Texture Representation and Metric," *Comp. Vis. Graph. Im. Proc.*, Vol 23, pp. 187-206, 1983.

[88] Raafat, H. M. and Wong, A. K. C., "A Texture Information-Directed Region Growing Algorithm for Image Segmentation and Region Classification," *Comp. Vis. Graph. Im. Proc.*, Vol. 43, pp. 1-21, 1988.

[89] Gagalowicz, A., "Visual Discrimination of Stochastic Color Texture Fields," in *Signal Processing: Theories and Applications* (eds. Kunt, M. and de Coulon, F.), pp. 289-295, 1980.

[90] Hawkins, J. K., "Textural properties for pattern recognition," in *Picture Processing and Psychopictorics* (eds. Lipkin, B. S. and Rosenfeld, A.), pp. 347-370, Academic Press, New York 1969.

[91] Zucker, S. W., "Toward a Model of Texture," *Comp. Graph. Im. Proc.*, Vol. 5, pp. 190-202, 1976.

[92] Zucker, S. W., Rosenfeld, A. and Davis, L. S., "Picture segmentation by texture discrimination," *IEEE Trans. Comput.*, Vol. C-24, pp. 1228-1233, 1975.

[93] Ahuja, N. and Rosenfeld, A., "Mosaic Models for Textures," *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol. PAMI-3, No. 1, pp. 1-11, January 1981.

[94] Wang, S., Dias Velasco, F. R., Wu, A. and Rosenfeld, A., "Relative Effectiveness of Selected Texture Primitive Statistics for Texture Discrimination," *IEEE Trans Sys. Man Cybern.*, Vol. SMC-11, No. 5, pp. 360-370, May 1981.

[95] Gonzalez, R. C. and Wintz, P., *Digital Image Processing (2nd Ed.)*, Addison-Wesley, 1987.

[96] Burt, P. J., Hong, T-H. and Rosenfeld, A., "Segmentation and Estimation of Image Region Properties Through Cooperative Hierarchical Computation," *IEEE Trans. Sys. Man Cybern.*, Vol. SMC-11, No. 12, pp. 802-809, December 1981.

[97] Spann, M., "Texture Description and Segmentation in Image Processing," *PhD. Thesis, University of Aston in Birmingham*, September 1985.

[98] Spann, M. and Wilson, R., "A Quad-Tree Approach to Image Segmentation which Combines Statistical and Spatial Information," *Patt. Recog.*, Vol. 18, Nos. 3/4, pp. 257-269, 1985.

[99] Volet, P., "Analyse et Synthese D'Images de Textures Structurees," *Ph.D Thesis, Ecole Polytechnique Federale de Lausanne*, 1987.

[100] Faber, T. L. and Stokely, E. M., "Orientation of 3-D Structures in Medical Images," *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol. PAMI-10, No. 5, pp. 626-633, September 1988.

[101] Cyganski, D. and Orr, J. A., "Applications of Tensor Theory to Object Recognition and Orientation Determination," *IEEE Trans. Patt. Anal. Mach Intell.*, Vol. PAMI-7, No. 6, pp. 662-673, November 1985.

[102] Franke, U., "Selective Deconvolution: A New Approach to Extrapolation and Spectral Analysis of Discrete Signals," *Proc. ICASSP*, pp. 1300-1303, Dallas, TX, April 1987.

[103] Franke, U., Meister, R. and Aach, T., "Constrained Iterative Restoration Techniques: A Powerful Tool in Region Oriented Texture Coding," *Proc. EUSIPCO*, pp. 1145-1148, Grenoble, France, September 1988.

[104] Franke, U. and Meister, R., "Region based image representation with variable reconstruction quality," *SPIE Vol. 1001 Vis. Commun. Im. Proc.*, pp. 178-186, Cambridge, MA, November 1988.

[105] Deguchi, K. and Morishta, I., "Texture Characterization and Texture-Based Image Partitioning Using Two-Dimensional Linear Estimation Techniques," *IEEE Trans. Comput.*, Vol. C-27, No. 8, pp. 739-745, August 1978.

[106] Stanger, V. Private Communication.

[107] Fejes Toth, L., *Regular Figures*, Pergammon Press, 1964.

[108] Lord, A. E. and Wilson, C. B., *The Mathematical Description of Shape and Form*, Ellis Horwood, Chichester 1986.

[109] Peli, E. and Goldstein, B., "Contrast in images," *SPIE Vol. 1001 Vis. Commun. Im. Proc.*, pp. 521-528, Cambridge, MA, November 1988.

[110] Pearson, D. E., "Model-based image coding," *Proc. IEEE Conf. Global Telecoms.*, Vol. 1, pp. 554-558, Dallas, TX, November 1989.

[111] Walsh, W. J., "Model-based coding of videophone images," *IEE Electon. Commun. Eng, J.*, Vol. 3, No. 1, pp. 29-36, February 1991.

[112] Limb, J. O., "Distortion Criteria of the Human Viewer," *IEEE Trans. Sys. Man Cybern.*, Vol. SMC-9, No. 12, pp 778-793, December 1979.

Appendix A. Details of Experimental Viewing Conditions and Generation of the Colour Prints

The viewing conditions under which all colour images were evaluated were as follows :

1. The monitor used was a 625 line interlaced (at 50 hz), 20 inch diagonal RGB monitor (screen height was 11.5"; screen width was 15"). The monitor was fed from a frame store such that a 512x512 image occupied the entire screen. All the images shown in this thesis are 256x256. The monitor gamma is unknown, and was assumed to be between 2.6 and 2.8, i.e. approximately equal to *e*.

2. The images were viewed in a large room which was normally well lit, but which could be darkened by the use of drawn curtains and turning off the lights. Viewing distance was typically four to eight times screen height.

The colour photographs provided throughout this thesis were obtained under the following conditions, which differ slightly from those above :

1. Each digital colour image was photographed manually from the display of a Sun workstation in a darkened room. The video source was in fact the output from an on-board TAAC-1 graphics accelerator. Each displayed image measured approximately 3" x 3".

2. The film used was ASA 200, colour negative, C-41 process. The camera f-stop was set at 8 and each exposure lasted one second.

3. The negatives were developed at a professional processing laboratory and initially contact processed for exposure check. Each negative was then individually custom printed, to obtain a good colour and density match, on 4" x 5" semi-matte paper. All colour and density matches were made by the laboratory staff.

The resulting photographs, measuring approximately 3.5" x 3.5" after cropping, are similar in size to the images displayed on the workstation. It is recognised that with the photographic method described above, some colour distortion and specular artefact may be introduced into the hardcopy. Because these photographs are placed on a bright surround, the white paper of the thesis, the apparent contrast will also be somewhat different from that of an image on a video display. However, the photographs in this thesis resemble the images displayed on the workstation as closely as I could achieve. The considerable cost of obtaining an exact colour and density match in each print prohibited the use of such a process.