

# **Algorithmic and learning based filtering techniques with application to colour image noise suppression and enhancement**

A thesis submitted for the degree of

Doctor of Philosophy

of the University of London

by

Derek Robin Charles



Machine Vision Group, Department of Physics,  
Royal Holloway, University of London

2003

## **Abstract**

This thesis investigates the principal local-window noise-removal techniques in the spatial domain using linear and non-linear filters for a variety of colour image representations and general and specialized image classes.

The thesis starts by describing the nature of colour, general image processing capture and storage techniques, and the manner in which image contamination can occur. A software framework was designed and implemented in which the image filtering algorithms were developed: its architecture and capabilities are outlined.

After a review of the current state-of-the-art in the field of vector filtering, performance comparisons of well known nonlinear and hybrid filters are achieved using established noise metrics; the validity and efficacy of these noise metrics are also debated. A number of new filters based on the vector median filter are introduced with a discussion of their relative merits. The new distance-weighted median filter is shown to be superior in its noise suppression performance characteristics using the NMSE criterion.

Particular regard is given to an extension of the truncated-median filter of Davies (1988), so that it can cope with colour images, with a discussion on its performance with highly contaminated images and a comparison of the distortions introduced by the new filter and those of the vector median filter. The new version of the truncated median filter is shown to be quite remarkable in its capability for extraction of partially hidden patterns in colour images.

The performance of the vector median filter and the variants described are contrasted with the capabilities of artificial neural networks (ANNs), in particular the multilayer perceptron, and the applicability of ANNs to colour image filtering is compared with previously published work on their use in greylevel images.

The thesis contains a considerable amount of review of previous work in the subject area, together with a substantial list of references.

# Acknowledgements

I would like to express my gratitude to the following people:

My supervisor Prof. Roy Davies for his support, direction and advice throughout my time at Royal Holloway.

My friend and postgraduate colleague, Tim Hampshire, who kindly provided me with sample bitmap loading and display code, and without whose help and advice my programming skills would have taken much longer to develop.

My friends and colleagues at PurpleVoice Ltd., their understanding and flexible approach to my working hours whilst writing this thesis have been invaluable.

My friend Peter Stevens, for his generous help and time in providing me with quality digital images for use in this work.

My partner Helen, whose patient love and encouragement played no small part in helping me complete this thesis.

I declare that the work presented in this thesis is my own.

This thesis is dedicated to my late father, Derek Richard Charles (1923 – 1997)

# Contents

Abstract .....	2
Acknowledgements .....	3
Contents.....	4
Chapter 1: Introduction .....	9
1.1 Introduction to machine vision.....	9
1.2 The phenomenon of colour (human visual system) .....	10
1.3 Motivation for colour usage in machine vision.....	13
1.4 Image capture .....	15
1.5 Image encoding .....	17
1.5.1 Lossless compression .....	18
1.5.2 Lossy compression .....	20
1.6 Image contamination .....	24
1.7 Requirement for image filtering.....	25
1.8 Aim of thesis .....	26
1.9 Thesis structure .....	27
Chapter 2: Colour Image Representation .....	28
2.1 What is a colourspace? .....	28
2.2 RGB.....	29
2.3 Ohta .....	30
2.4 HSI-type colourspaces.....	32
2.5 YCbCr .....	35
2.6 CIE XYZ .....	36
2.7 CIE LUV .....	38
2.8 CIE LAB .....	39
2.9 Concluding remarks .....	40
Chapter 3: The Colour Image Processing Environment.....	41
3.1 Image processing environment requirements.....	41
3.1.1 Use of resources .....	41



3.1.2 Multiple image display .....	41
3.1.3 Storage format compatibility.....	42
3.1.4 Image size.....	42
3.1.5 Operability.....	42
3.1.6 Real-time applicability .....	42
3.2 Available software.....	42
3.3 Implementation.....	43
3.3.1 Functionality.....	45
3.4 Test images used in this thesis .....	47
3.5 Concluding remarks .....	56
Chapter 4: Local Window Methods of Noise Removal .....	57
4.1 Definition of the local window.....	57
4.2 Convolution operators .....	59
4.2.1 The Mean Filter .....	60
4.3 The Median Filter.....	61
4.4 The Vector Median Filter.....	63
4.4.1 Definition 1: Deriving the vector median using rank order statistics ....	64
4.4.2 Definition 2: Deriving the vector median as a maximum likelihood estimator .....	66
4.5 Other vector filters .....	67
4.5.1 The $\alpha$ -trimmed Vector Median Filter ( $\alpha$ VMF) .....	68
4.5.2 The Basic Vector Directional Filter (BVDF).....	68
4.5.3 The Generalized Vector Direction Filter (GVDF).....	69
4.5.4 The Directional Distance Filter (DDF) .....	70
4.5.5 The Directional Magnitude Vector Filter (DMVF).....	70
4.5.6 Adaptive vector filters.....	70
4.6 Noise measurement metrics .....	72
4.7 Performance of discussed filters .....	76
4.7.1 Experiment 1: 2% random impulse noise .....	76
4.7.2 Experiment 2: 5% random impulse noise .....	79
4.7.3 Experiment 3: light Gaussian noise ( $\sigma^2=100$ ) .....	81

4.7.4 Experiment 4: moderate Gaussian noise ( $\sigma^2=1000$ ).....	83
4.7.5 Experiment 5: light mixed noise (2% + $\sigma^2=100$ ) .....	85
4.7.6 Experiment 6: moderate mixed noise (5% + $\sigma^2=1000$ ).....	87
4.7.7 Discussion .....	89
4.8 Artificial Neural Networks.....	91
4.8.1 Application to noise removal .....	91
4.8.1 The perceptron.....	92
4.8.2 The multilayer back-propagation perceptron .....	93
4.8.3 Other network topologies .....	96
4.9 Concluding remarks .....	96
Chapter 5: Extensions to the Vector Median Filter.....	97
5.1 Experiments undertaken.....	97
5.2 Vector median filtering in other colourspace .....	97
5.2.1 Results .....	97
5.2.2 Concluding remarks .....	101
5.3 Hybrid Mean Median Filter (HMMF).....	101
5.3.1 Results .....	105
5.3.2 Concluding remarks .....	105
5.4 Selective Rank Order Filter (SROF) .....	106
5.4.1 Results .....	106
5.4.2 Concluding remarks .....	106
5.5 Distance Weighted Vector Median Filter (DWVMF).....	111
5.5.1 The rationale of the new filter .....	112
5.5.2 Results .....	115
5.5.3 Discussion .....	126
5.5.4 Concluding remarks .....	127
5.6 Fast calculation of the VMF .....	131
5.6.1 Implementation.....	133
5.6.2 Results and discussion.....	134
5.7 Concluding remarks .....	135

Chapter 6: Mode Filtering .....	137
6.1 Introduction to mode filtering and the Truncated Vector Median Filter.....	137
6.2 Problems of mode filtering and a solution .....	138
6.3 Methodology for applying mode filtering to colour images .....	142
6.4 Application of mode filters to colour images.....	146
6.4.1 Qualitative results.....	146
6.4.2 Edge enhancement.....	149
6.4.3 Location of mode .....	149
6.4.4 Noise removal .....	156
6.5 Edge shifting characteristics of mode filters .....	160
6.5.1 Case 1 .....	168
6.5.2 Case 2 .....	168
6.5.3 Case 3 .....	172
6.5.4 Case 4 .....	173
6.6 Finite edge width model: calculation of breakpoints .....	174
6.7 Concluding remarks .....	180
Chapter 7: Application of Neural Networks to Colour Image Filtering .....	182
7.1 Application to greyscale images .....	182
7.2 Architecture of MLP .....	183
7.3 Training set.....	184
7.3.1 Standard images .....	186
7.3.2 Plastic component images .....	186
7.3.3 Confectionery images.....	188
7.4 Experimental results.....	188
7.4.1 Training .....	191
7.4.2 Testing.....	194
7.4.3 Comment on colour space suitability .....	195
7.4.4 Comparison with conventional filters .....	199
7.4.5 Combining neural network with nearest neighbour selection.....	203
7.5 Concluding remarks .....	205

Chapter 8: Conclusions .....	211
8.1 Review of thesis .....	211
8.2 Future work .....	214
Appendix A .....	216
A1 CIE definitions .....	216
A2 Television colour spaces .....	217
A3 Conversion between RGB standards .....	220
Appendix B .....	221
Appendix C .....	222
C1 ANN training results .....	222
C2 ANN testing results .....	222
C3 Comparison with conventional filters .....	235
Appendix D .....	242
Bibliography .....	244

# Chapter 1: Introduction

## 1.1 INTRODUCTION TO MACHINE VISION

Digital image processing is still a relatively young area of physics that has become possible, and indeed advanced, with the development of digital computers. Images had been coded digitally as early as 1921 using telegraph printers with special typefaces to reproduce pictures with up to five distinguishable levels of brightness. However, it was not until the 1960's that image processing became a science in its own right, and the needs of the NASA space program required that serious work be done on methods to enhance received picture data from early space probes.<sup>1</sup>

Since then, huge advances in computer technology have made more complex algorithms feasible in practice, and applications for image processing techniques have been found in a wide variety of fields, such as facial recognition, satellite imagery, fingerprint identification, food inspection, car speed measurement, X-ray enhancement, aircraft identification and tracking, and astronomy. The list is almost endless.

In fact, it is difficult to think of any discipline that involves visual representation of information which would not benefit from image processing in one way or another. Thus it has rapidly become a highly important subject area, and although extensive research has been done in the last quarter century, there is still much scope to develop more advanced processing techniques for applications old and new.

One of the recent new areas for investigation has been the field of colour image processing. The following sections enlarge on the basis of colour imaging and its application to the field of machine vision.

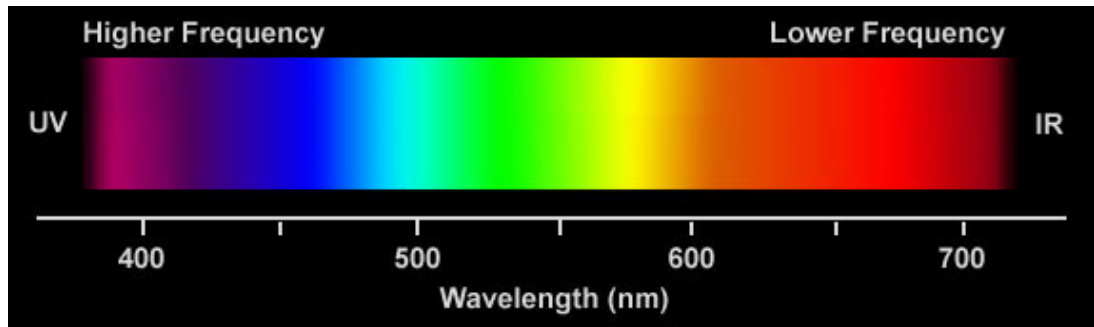
## 1.2 THE PHENOMENON OF COLOUR (HUMAN VISUAL SYSTEM)

The eye is a highly evolved organ; the detection of the presence or absence of light is essential for virtually all animals. The structure of the eye may vary between species according to specific needs, but the common requirement to find food, seek shelter, or avoid danger through analyzing the surroundings remains the same. In those species where accurate spatial location is highly important (e.g. predatory birds such as eagles), both eyes have evolved to be forward-facing to permit excellent depth/distance judgment.

One common factor across all species is the discrimination of light change and variation, regardless of the wavelength. In the human vision system, the “rod” receptors (so-called because of their shape) react to changes in light intensity but not wavelength, and are therefore not involved in colour vision. They are sensitive to low levels of illumination, and are thus suited to night vision. There are some 75 to 150 million rods in the eye, but because several rods are usually connected to a single nerve ending, detail resolution is poor. The other class of receptor found in the human eye is the “cone”, and there are three different types – S-, M-, and L-cones, classed according to their spectral sensitivities (Short, Medium and Long wavelengths). There are six to seven million cones in the eye, primarily located in the fovea (the central part of the retina) where our vision is best. Cones are highly sensitive to chromatic changes, and with them humans are capable of picking out fine detail, largely because each cone has its own nerve connection.<sup>1</sup>

The spectral coverage of the S-, M- and L-cones broadly corresponds to the sensations of red, green and blue hues (Fig. 1.2a).<sup>4</sup> Mean values for the wavelengths of maximal absorption are 426nm for the blue pigment, 530nm for the green pigment, and 552 nm and 557nm for two polymorphic variants of the red pigment.<sup>2</sup> The human eye has a spectral response ranging from approximately 380nm to 710nm, shown in Figure 1.1.

Trichromacy (that is, having three photoreceptors with different spectral characteristics) is a physical attribute we share only with other primates. Other



**Figure 1.1. Frequency range of the human visual system (www.photo.net).**

mammals including New World monkeys are dichromats, having an S pigment and a single shared L/M pigment.<sup>3</sup> Separate L and M pigments have evolved fairly recently, in the order of 30 million years ago, as a result of a duplication on the X chromosome. This is borne out by the fact that the L and M genes lie next to each other on the X chromosome, and are 96% identical.<sup>4</sup> One can see the extensive overlap between the two cone types in Figure 1.2a, and it is thought that this second colour subsystem may have evolved to allow differentiation between reddish and greenish colours, to allow distinction between fruit and foliage.<sup>5</sup>

Figure 1.2b gives the normalized excitations for the L and M cones, showing that although the peak sensitivities are only about 30nm apart (Figure 1.2a), the ratio of excitation in this region provides unambiguous information for distinguishing between two lights even of similar wavelength, enabling us to be extremely discriminatory with respect to both chrominance and luminance information in this frequency band. Conversely, because the peak sensitivity of the S-cones is at a much shorter wavelength than that of the M- or L-cones, coupled with the fact that S-cones comprise less than 10% of the total number of cones in the eye, it has been shown that they primarily contribute chromatic information.<sup>4,6</sup>

Figure 1.2c shows the excitation space of all three cones, and any viewable light source can be represented as a point within this space.

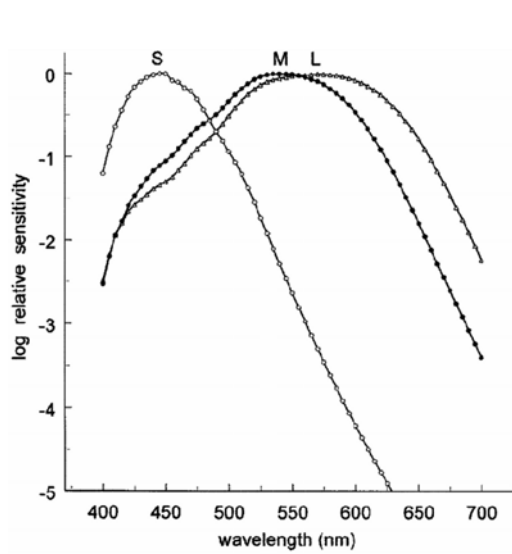


Figure 1.2a. Sensitivities of the different types of cone in the human eye (Webster, 1996).

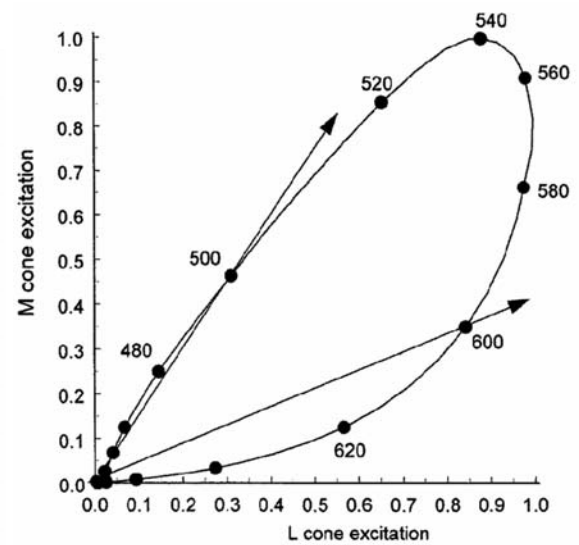


Figure 1.2b. Normalized excitations of M and L cones (Webster, 1996).

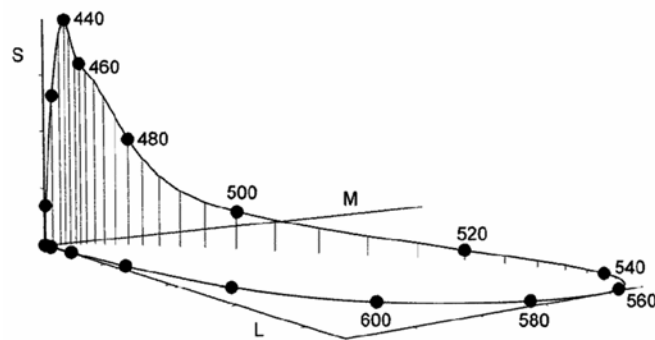


Figure 1.2c. Excitation space and sensitivity of the three cone types (Webster, 1996).

It is important to note that having only three cone types restricts our perception of colour to only three colour dimensions, and if any two lights are matched to produce equivalent photon absorption, any information about their difference is lost, regardless of any post-processing.<sup>4</sup>

However, it is precisely this limitation which allows us to record or reproduce full-colour images artificially (e.g. television) using only three primary light sources of different wavelengths, with the restriction that no primary can be reproduced by any combination of the other two. It has been shown that specific values for red, green and blue frequencies can give a greater range of discernible colours than no other



combination can.<sup>7</sup> Due to the eye's non-linear perception of colour, this does not mean that all colours can be obtained from a combination of these primaries.

### **1.3 MOTIVATION FOR COLOUR USAGE IN MACHINE VISION**

Opinion is divided on whether chrominance information is especially useful or even necessary compared to luminance when processing or interpreting images. As has been stated previously, many species do not have the ability to discriminate colours, so can colour be that vital?

Several psychophysical studies suggest that colour is unimportant or not discernible for a wide range of human visual activities, ranging from detecting the colour of a moving object<sup>8</sup>, obtaining shape from shading, or illusory contours,<sup>9,10,11</sup> and that moving chromatic stimuli produce the same response as low-contrast luminance stimuli.<sup>12</sup> McIlhagga et al. found little difference in chrominance and luminance perception when varying the contrasts between the two during experiments with low spatial frequency textures.<sup>13</sup>

On the contrary, Healey and Slater have found that the performance of feature matching algorithms improves when using colour because it increases the amount of information available for a given region, and is thus highly important for the recognition of small objects.<sup>24</sup>

It is worth emphasizing that all mammals have a chrominance component to their vision, so it must be useful in some respect; it is probably no coincidence that certain animals have developed coloured coatings to take advantage of this, such as the distinctive yellows of bees and wasps, and the reds of butterflies, beetles and ladybirds, to warn off predators. It would seem that colour vision is a continuing evolution of an already complex visual system designed to extract the maximum information from a scene if there is sufficient computational power to process it. Indeed, the great majority of image processing and machine vision papers published refer only to binary or grey-level images, as some workers have noted.<sup>14,15,16</sup> The most obvious reason for this is the great expense, both in terms of equipment needed

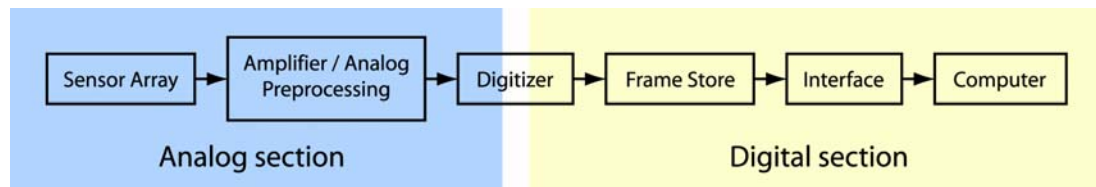
to acquire and display images, and the additional processing required, when dealing with colour information.<sup>17,18</sup>

Whilst this was perhaps true five or ten years ago, hardware capable of capturing and displaying full 24-bit colour images has dramatically fallen in price, and similarly computational power has vastly increased. For example, the MMX and SSE capabilities of recent AMD and Intel processors were designed with signal processing operations in mind, and the instruction sets have been optimized for fast addition, multiplication and matrix operations.<sup>25,26,27</sup>

With the onus on use of minimum memory and simple, fast algorithms somewhat relaxed as a result, more research has been focused on applications that necessitate colour discrimination. Areas include vegetable ripeness and/or toxicity<sup>16,19</sup>, leaf decay<sup>16</sup>, geological classification<sup>20</sup>, natural scene classification<sup>15,21</sup>, computer aided fashion design<sup>22</sup>, and bruise detection on fruit.<sup>23</sup>

One discipline in particular in which colour image processing and analysis is of prime importance is that of medical physics, ranging from pharmaceutical blister pack inspection<sup>31</sup>, wound metrics<sup>28,29</sup> to traditional Chinese medicine<sup>30</sup>. It has been shown in these publications that accurate colour information is vital in this area, and it is expected that considerable advances in colour image analysis will be made in this field.

On a more commercial level, the great boom in electronic publishing (especially on the World Wide Web) has spawned a multitude of methods for image compression and image registration, and the availability of a plethora of low-cost colour imaging devices such as scanners and digital cameras means that colour image processing techniques are used by a greater number of people – notably more home users – than ever before.



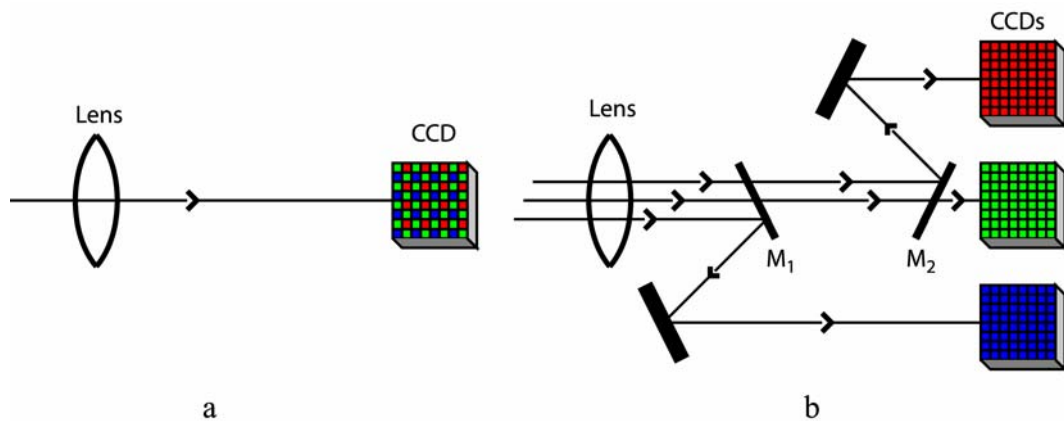
**Figure 1.3. Schematic diagram showing image capture path.**

In summary: just as nature has evolved colour vision, so the evolution of computational hardware enables full-colour image acquisition and display. As the availability of such hardware has increased, new and varied applications have rapidly been found which necessitate some form of post-processing – and so a worthwhile area for research has opened for more widespread scrutiny and investigation.

## 1.4 IMAGE CAPTURE

All modern capture equipment, be it scanner, camera, or specialized device, can be considered to consist of five components, as shown in Fig 1.3.

The mechanics of the sensor array will clearly differ according to the application, but high quality and wide availability of CCD (Charge-Coupled Device) sensors make them the most commonly found acquisition devices for visible light images. A CCD consists of an array or matrix of metal oxide semiconductor material made up from tessellating elements, such as hexagons or squares. Charge is generated in each element when light falls upon it, proportionate to the level of incident radiation. The discrete element approach of the CCD has an inherent advantage over line-scan cameras, in that the entire image can be sampled at once rather than in a raster scan fashion; this eliminates any discrepancies due to scene motion over the time taken to acquire the image, and means the captured image is already spatially quantized. However, the output is still an analogue signal which requires digitization before transfer to image processing hardware. A further advantage in using CCDs is the high sensitivity of modern units; even single photons may be detected.

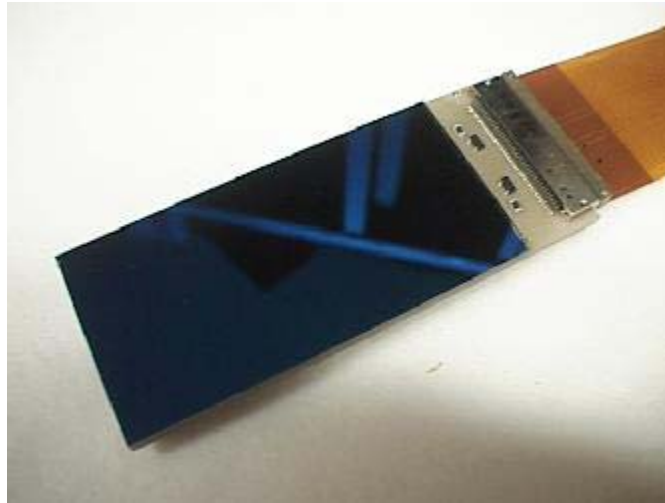


**Figure 1.4. Colour image capture using 1- and 3-CCD arrangements.**

To acquire colour images, CCDs may be employed in two different ways. The first configuration only utilizes one CCD matrix, in front of which a colour filter grid is placed. The colour of an individual element is calculated from the averaged response of it and its neighbours. The pattern most often used is G-R-G-B, called the Bayer pattern after Bryce Bayer of Kodak, shown in Figure 1.4a. It is biased toward producing accurate green colours because the eye is more sensitive to those wavelengths. Half the CCD elements measure green, while a quarter measure red, and the other quarter measure blue. Another pattern used is a vertical arrangement of red, green and blue striped filters, much like a shadowmask in a colour television CRT (Cathode Ray Tube). All pixels, taken in order, are used for luminance. Every third pixel represents red, green or blue, depending on where sampling begins. The fact that the luminance signal is sequential RGB samples rather than a true monochrome signal in addition to the fact that any single colour only has one-third of the luminance resolution limits the colour quality of single chip devices. A more subtle problem is that any one primary colour is not sensed over the entire surface of the CCD, like looking at the subject through a vertical Venetian blind. A subject with vertical bands of color may have significant chromatic distortion.

Other patterns are sometimes used, and Canon has developed a filter pattern of cyan, yellow, green and magenta (for their still digital cameras).

The other configuration is three separate CCDs producing a proper tristimulus output, with red, green and blue filters in front of each respective CCD (Figure



**Figure 1.5. Example of  $2048 \times 4096$  resolution CCD, actual size 3.5cm by 7.0cm (MIT Lincoln Laboratories).**

---

1.4b). The incoming light is separated by dichroic mirrors ( $M_1$  and  $M_2$ ) which according to their thickness pass the light to the appropriate filter. Since each CCD produces only one colour, all components of the signal are at full system resolution; also, little light is lost, resulting in truer hues and greater sensitivity. The penalty is cost, size, and weight.

CCDs in domestic still and video cameras have areas in the region of a few square centimetres, giving a resolution in the low millions of pixels. CCDs used in professional applications can be somewhat larger, with tens of thousands of elements along each dimension. Figure 1.5 shows a CCD used for astronomical image capture at the MIT Lincoln Laboratories.

## 1.5 IMAGE ENCODING

Images are represented as a square array of elements, known as a pixels, with the horizontal and vertical axes represented by  $x$  and  $y$ . It is arbitrary whether the  $y$  origin is at the top or bottom of the image, but in this text the convention of  $y$  originating at the top is adopted. A digitized image can therefore be considered as a two dimensional function  $f(x,y)$ . Greyscale images simply encode intensity, and thus consist of only scalar quantities; colour images are represented by a number of

discrete components, so each pixel contains a vector corresponding to its position in the particular colour mapping scheme used. Whatever colour mapping system is used, the final display unit output is characterized in terms of intensities of red, green and blue phosphors. The amount of storage required per pixel depends on the resolution of the analogue-to-digital converter (ADC) used when acquiring the image; commonly, 8 bits per colour channel are used, or 24 bits per pixel, allowing up to  $2^{24}$  (16.7 million) different colours to be displayed. Although the eye is only capable of discriminating between some seven million different hues, its response is highly nonlinear, and only 8 bits per channel can lead in some cases to coarsely represented images. In professional applications 48 bit (16 bits per channel) formats are becoming more common.<sup>32</sup>

It can easily be deduced from this that colour image storage requirements are quite significant for any application requiring greater than a medium-resolution image. For instance, the storage required for a  $1024 \times 1024$ , 24 bit image is well over 3MB. Unless storage or transmission bandwidth is not an issue, some form of compression is employed to encode the image more efficiently.

### **1.5.1 LOSSLESS COMPRESSION**

Lossless compression schemes exploit the *data* redundancy present in the image, but retain all the informational content of the original image. Repeated encoding and decoding cycles do not result in degeneration. Lossless compression methods yield typical compression ratios in the order of 2:1. Many different schemes have arisen, for instance:

#### **RUN-LENGTH CODING**

It is common to find groups of pixels of the same value occurring contiguously; instead of representing each pixel value discretely, a component pair encodes the number of pixels to be repeated and their group value. This scheme is generally used

with coarsely quantized, low spatial frequency images; otherwise the storage penalty may well be higher than the raw 24-bit representation.

## **DICTIONARY CODING**

The premise of dictionary coding is the assignment of fixed length codes to variable lengths of pixels. Each dictionary entry may refer to other dictionary codes already encoded. Originally invented in 1977 by Ziv and Lempel, with later improvements by Welch, typical compression ratios of 33% are achieved. The completed dictionary is stored with the coded data. Encoding only requires one pass over the data; it requires no prior statistics. It is commonly used in the encoding of TIFF (Tagged Image File Format) images.<sup>28,127,128,129</sup>

## **ENTROPY CODING**

As a result of Shannon's work on communications theory, it is well known that for a given communication channel the lower bound on the average bit rate of a sequence cannot be less than its entropy  $H$ , defined as<sup>33</sup>:

$$H = -K \sum_{i=1}^n p_i \log p_i \quad (1.1)$$

Where  $p_i$  is the probability of an occurrence of symbol  $i$ .  $K$  is an arbitrary scaling constant. Entropy coders seek to get close to this Shannon limit by exploiting the statistical nature of the pixels to be encoded; commonly occurring values are given shorter codes. Clearly this method works best when the input symbol probability distribution is non-uniform. Unlike dictionary coding and run-length coding, this method requires two passes over the data set; one to ascertain the symbol probabilities, the other to encode the data. The source and receiver may both use predefined code tables (such as in facsimile transmission), or separate code tables may be transmitted for each image. The former saves on transmission bandwidth but the coding efficiency may be sub-optimal as the particular code probabilities of each

image are not exploited; the converse is true for the latter. Huffman and Gray encoding are well-known methods of entropy coding.

## 1.5.2 LOSSY COMPRESSION

As the term implies, lossy compression involves discarding information from the image irreversibly. Generally, the way in which the removal of data occurs is decided according to psycho-visual principles to minimize the apparent loss of visual quality. The reconstructed image after decompression is thus an approximation of the original image. Lossy compression schemes involve transformation of the data into different domains with subsequent quantization and/or thresholding of the resulting data.

Lossy schemes can achieve much greater data reduction than lossless schemes, typically in the order of 100:1.

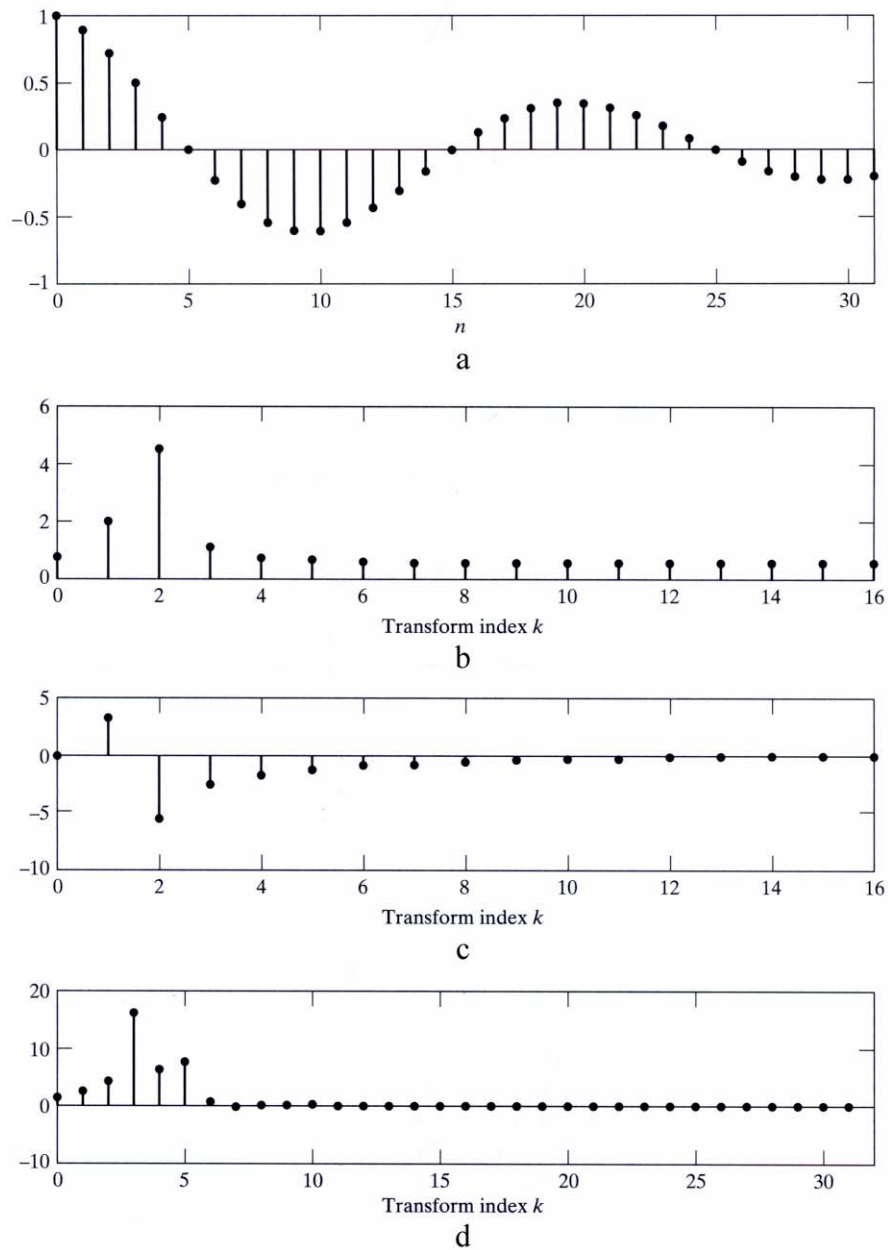
## PRINCIPAL TRANSFORMATIONS

The most well known transform from the spatial domain in digital image processing is possibly the Discrete Fourier Transform (DFT), given in two dimensions by:

$$H_{n,m} \equiv \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} h_{l,k} e^{2\pi i k m / M} e^{2\pi i l n / N} \quad (1.2)$$

Whilst having been proven effective for texture analysis<sup>36</sup>, its properties are less than ideal for energy compaction; the non-zero transform coefficients are widely spread. However, by transforming from the spatial domain using only cosine components using the DCT (Discrete Cosine Transform), over 90% of the signal energy is concentrated in only a few coefficients; the contribution of the higher frequency components is negligible; Figure 1.6 shows a typical signal and the first 32 coefficients of the resulting DFT and DCT. As Figure 1.7 shows, the reconstruction error using truncated coefficients is far lower for the DCT than for the DFT. The energy compaction and sparse representation properties of the DCT are very close to

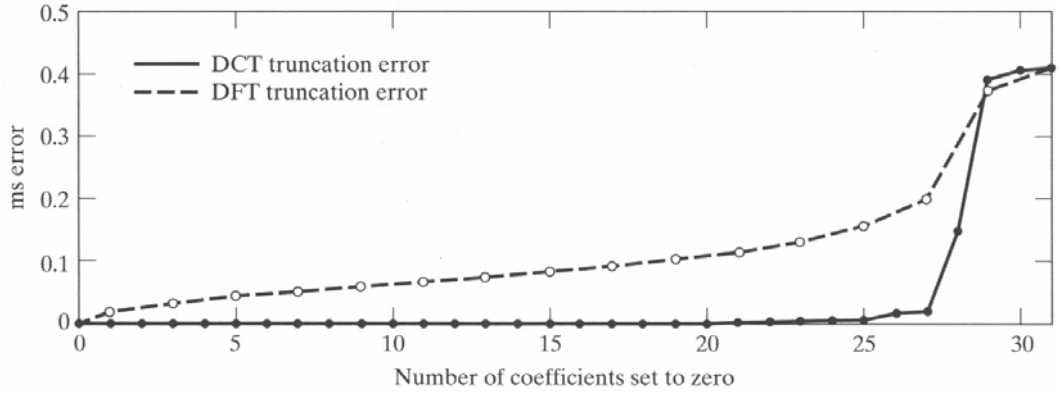




**Figure 1.6. Typical sinusoidal digital signal (a) with the first 32 real (b) and imaginary (c) DFT coefficients compared to (d) first 32 DCT coefficients (Oppenheim & Schafer, 1999).**

those of the optimal Karhunen-Loeve Transform (KLT), also known as principal components analysis<sup>17</sup>, without the need for computationally expensive data analysis and matrix diagonalization for individual images.

The two dimensional DCT is given by:



**Figure 1.7. Comparison of truncation errors for DFT and DCT (Oppenheim & Schaffer, 1999).**

$$F_{n,m} = \frac{1}{4} C_n C_m \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} f_{j,k} \cos \left[ \frac{\pi(2j+1)m}{N+M} \right] \cos \left[ \frac{\pi(2k+1)n}{N+M} \right] \quad (1.3)$$

where

$$C_z = \begin{cases} 1/\sqrt{2} & z = 0 \\ 1 & z \neq 0 \end{cases}$$

The DCT is used as the transform kernel in both JPEG still image compression and MPEG moving picture compression. In both cases the source image (or frame) is subdivided into  $8 \times 8$  blocks and the first  $K$  transform coefficients are quantized and Huffman coded.  $K$  is dependent on the level of compression required. The image subdivision approach and the bias toward keeping only low-frequency components can lead to poor performance at object edges, and a distinct “blockiness” to reconstructed images.

One of the deficiencies of the DFT and DCT is their lack of space-frequency resolution; they operate on the basis that the signal or image to be analyzed is periodic. To combat this problem the notion of ‘windowing’ was introduced; however this has its own problems with ringing and discontinuities at the edge of the support. An alternative transform method which addresses the space-frequency

resolution problem, and which underpins the new JPEG2000 standard<sup>35</sup>, is the Discrete Wavelet Transform (DWT).

The DWT uses two orthonormal wavelet bases with compact support to decompose an image into low- and high-frequency filtered versions. The wavelet bases in discrete form are essentially FIR (Finite Impulse Response) filters. The high-frequency information is stored, and the low-frequency result is decimated by two before the DWT is reapplied. This process is then repeated until the data to be processed is of length 1. The process may be viewed as a binary tree in which only one node at any branch has children. A generalization of the DWT is the Wavelet Packet Transform, in which many decomposition tree variations are permitted. This approach has the advantage that either detailed frequency or phase information may be extracted from a given frequency sub-band depending to what depth the decomposition tree is traversed. Conversely, the DFT or DCT only allow detailed time resolution at the expense of frequency, and vice versa – hence the block-based approach of the JPEG compression method.

The DWT or WPT, able to be applied to the image as a whole, do not suffer as badly from impulse ringing and block discontinuities: visually, a wavelet-compressed image is of higher quality, and thus a higher compression ratio can be achieved for a similar visual standard to the original JPEG.

The subject of image compression using wavelets is thoroughly discussed in a recent thesis by Strømme.<sup>35</sup>

## **PRINCIPAL QUANTIZATION METHODS**

As has already been discussed, quantization of transformed image coefficients improves the data reduction. However, it may be desirable in some situations to quantize image pixels directly. For instance, it is likely that an image will only contain a small proportion of the whole gamut of colours available for display – by limiting the number of possible symbols, more efficient entropy coding can be

employed. Or, there may be a requirement to display a reduced-palette image on video devices unable to show the original full range of hues.

Simple scalar quantization (SQ) merely reduces the number of bits available for each colour channel. Whilst computationally inexpensive, this can lead to false contouring and severe loss of detail in small objects.

A more effective approach is vector quantization (VQ). In this way, the colourspace of the image is subdivided into volumes of non-uniform size, and each pixel in that volume is mapped to a mean value. A palette codebook, similar to the Ziv-Lempel dictionary, is stored with the image after processing.

Many different image palette VQ techniques have been developed, and space precludes their discussion, but most notable are the  $k$ -means algorithm first developed by McQueen<sup>63</sup>, Heckbert's median cut<sup>54</sup>, the octree method<sup>67</sup>, and a recent promising system using a Kohonen, or self-organizing map, neural network<sup>68</sup>. The reader is directed to the references for a full treatment of the various merits of each system.

## 1.6 IMAGE CONTAMINATION

Noise and other unwanted artifacts in images can arise for a variety of reasons. Contamination may be introduced at any of the stages in Figure 1.3. In the analog section, noise may occur in the form of photon noise or dark current in the CCD, but these difficulties are easily avoided by increasing the capture integration time for the former, and Peltier cooling for the latter. Of more concern is generation of noise by the amplifier section prior to digitization. Amplifier noise is characterized as a Gaussian process, independent of the signal and additive. For a colour signal the noisy signal  $g$  is given by:

$$\begin{aligned} g_r(x, y) &= f_r(x, y) + n_r(x, y) \\ g_g(x, y) &= f_g(x, y) + n_g(x, y) \\ g_b(x, y) &= f_b(x, y) + n_b(x, y) \end{aligned} \tag{1.4}$$

where  $f$  is the original signal and  $n$  the additive Gaussian noise. The level of noise is given in terms of its statistical variance  $\sigma^2$ .

Signal-to-noise ratio (SNR) is unavoidably lowered in the digitization stage, by both the discrete spatial sampling and signal level quantization. The maximum SNR for a given sample point for a  $(N+1)$ -bit quantizer is<sup>37</sup>:

$$\text{SNR} = (6.02N + 10.8) \text{ dB} \quad (1.5)$$

In general, digital signals have better noise immunity than analog signals; however, once in the digital domain, artifacts may be introduced by an unreliable transmission channel. Should a single bit of a particular sample value be transmitted incorrectly the error in output value may be as high as 50%. This type of noise is entirely random and can be only typified in terms of the likelihood of correct transmission by the communications channel; the image contamination itself is characterized as impulsive noise in the midst of correct pixel values. The observation equation for impulsive noise can be written as<sup>73</sup>:

$$\begin{aligned} g_r &= ef_r + (1-e)z_r \\ g_g &= ef_g + (1-e)z_g \\ g_b &= ef_b + (1-e)z_b \end{aligned} \quad (1.6)$$

where  $g$  is the noisy signal,  $e$  indicates a random binomial variable taking the value 1 with a probability  $p$  (absence of noise) and a value of zero with a probability of  $1-p$  (presence of noise), and  $z$  is an independent random variable. The original image pixel  $f_{rgb}$  is replaced by the random colour pixel  $z_{rgb}$  with a probability  $1-p$ .

Other noise models may be described as Poisson, Cauchy, exponential, or uniform.

## 1.7 REQUIREMENT FOR IMAGE FILTERING

The presence of noise in an image can be disruptive to a wide variety of image processing methods; for compression it may alter the selection of coefficients, or increase their number to obtain a lower error metric; it may lead to false detection of

edges, and as edge detection is the premise for operations such as object recognition and region segmentation the ramifications may extend to a wide range of medium level techniques. In colour quantization, the erroneous data introduced by noise may lead to sub-optimal codebook or palette generation. In fact, any processing technique which relies on image statistics may be skewed by noise artifacts in the image.

In addition to removing the type of noise it was designed for, an ideal noise removal or enhancement filter should have the following characteristics:

- It should retain edges in the image without spatial shifting.
- It should not remove image detail or cause blurring.
- It should not introduce any chrominance or luminance distortion.

In practice this is not the case, and the filter must be chosen carefully depending on the noise model, the generality of its usage, and the salient information which will ultimately be extracted from the processed image.

## **1.8 AIM OF THESIS**

This thesis investigates principal noise-removal techniques in the spatial domain using linear and non-linear filters for a variety of colour image representations and general and specialized image classes. After comprehensive study of the current state-of-the-art in the field, performance of well-known non-linear and hybrid filters are thoroughly considered using established and recently developed noise metrics; the validity and efficacy of these noise metrics are also discussed. A number of new filters are introduced with a discussion of their relative merits. In the final chapter, previous work using neural networks in scalar-valued images is extended to their use in colour images and compared to the filters already considered.

Transform-based and morphological operators are not discussed, and are subjects worthy of study in their own right.

## **1.9 THESIS STRUCTURE**

Chapter 1 introduces the subject of colour image processing and its relevance to modern analysis and storage methods.

Chapter 2 outlines the various colour image representations in popular use, their evolution and their advantages and applications.

Chapter 3 discusses the software package developed to implement the algorithms, along with a discussion for the need of an independent environment for assessment of these methods. Also detailed are the images used in this thesis and the reasons for their choice.

Chapter 4 is a treatment of contemporary non-linear and hybrid filtering algorithms using local window methods, and also the noise metrics used to ascertain their efficacy and characteristics.

It also introduces the subject of neural networks and their application to image filtering.

Chapter 5 establishes a number of original extensions made to the vector median filter with comparisons to other well-known filters, along with a method for reduced complexity calculation.

Chapter 6 describes an extension based on the truncated-median filter of Davies, developed in a novel manner, with a discussion on the nature of highly contaminated images and an examination of the distortions introduced by the new filter and those of the vector median filter.

Chapter 7 presents original work using artificial neural networks applied to colour images. Again, results are presented and discussed in the light of filters treated in earlier chapters.

Chapter 8 discusses the work presented with concluding remarks, including possible areas for future work and enhancement.

# Chapter 2: Colour Image Representation

## 2.1 WHAT IS A COLOURSPACE?

A colourspace is a representation by which colours can be specified and created using parameters of discrete value. As colour is a purely sensory phenomenon the specification and measurement of colour is a highly complex subject: difficulties stem from the fact that the brain analyzes colour in a different manner than the eye cone response would indicate: we are not aware of colour as a mixture of multiple stimuli.

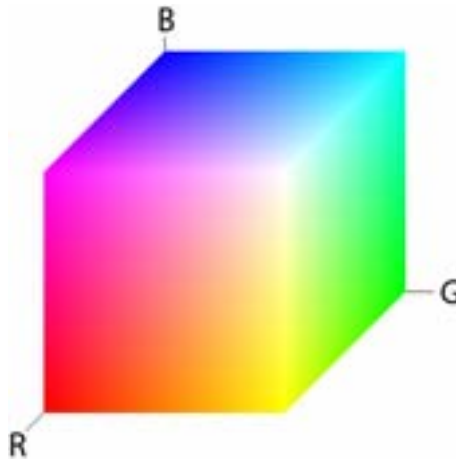
As there are three separate receptors in the eye, in general colourspaces similarly have three degrees of freedom. Two dimensional colourspaces have been used; results are adequate but are noticeably visually inferior<sup>38</sup>.

The *gamut* is the range of colours visible in the colourspace, and is usually the region bounded by the vertices of the primaries used.

Many colourspaces have been developed, arising from the different requirements of a given objective. Whilst it is not intended to give an exhaustive account of all methods currently in use, a broad selection of popular colourspaces is discussed with their application to colour image processing.

It should be noted that only *digital* colourspaces are described. Analogue television colourspaces are often used because of the ease of digitization of each channel direct from the capture camera. However, these colourspaces are pre-corrected for the non-linear transfer characteristics of the CRT and are tailored to analogue transmission and bandwidth limitations; in the interests of strict accuracy these should not be used for colour digital image processing. For the sake of completeness, a brief discussion and common transforms of television colourspaces are given in Appendix A2.





**Figure 2.1. The RGB colour cube.**

---

Due to the perceptual nature of colour, the CIE (Commission Internationale de l'Éclairage – International Commission on Illumination) has introduced strict definitions describing colour sensations with respect to physically measurable quantities. These definitions are given in Appendix A1.

## **2.2 RGB**

The RGB colourspace is the most well-known; each encoded channel corresponds to the output intensity of the red, green and blue display phosphors respectively. As many applications keep image data in native display format, the majority of conversions in the literature involve this colourspace.

The RGB gamut forms a cube with red, green and blue as orthogonal axes and the achromatic black-white axis as the lead cube diagonal, as shown in Figure 2.1. Black is given by  $R = G = B = 0$ , and white is  $R = G = B = 2^N - 1$ , when  $N$  is the bit depth. A monotonic increase in a given channel is matched by a linear increase in applied output.

The advantage of a display-based colourspace such as RGB is that all colours in the gamut are visible; there are no singularities or out-of-gamut colours. As a capture colourspace, RGB also has the advantage that any input noise will be uniformly

distributed and thus operations will be more robust<sup>48</sup>. However, there are a number of disadvantages:

- There is high correlation between the channels for natural images – an increase in colour brightness without a change in hue produces a corresponding rise across all three channels, i.e. intensity is a common factor.
- It is difficult to intuitively specify colour in terms of amounts of red, green and blue. Similarly, it is difficult to discern the overall hue of a pixel in this colourspace.
- Linear differences in between any RGB vector pair do not correspond to a linear perceptual difference.
- It is *device-dependent* – in other words, the same RGB encoded image may not look identical on two different displays.

With regard to the last point, recommended red, green and blue phosphor values have been specified by major standards organizations to permit somewhat more accurate conversion between RGB and other colourspaces. These bodies include the European Broadcasting Union (EBU), the Society of Motion Picture and Television Experts (SMPTE) and the International Television Union (ITU). Transforms between the common RGB colourspaces are given in Appendix A3.

## 2.3 OHTA

This colourspace was developed by Ohta<sup>38</sup> as a result of investigating effective colour features for image segmentation. The goal was to find colour axes which gave the largest discriminant power in feature selection, i.e. axes which had the greatest variance. This was achieved by applying the Karhunen-Loeve transformation to a set of natural colour images (originally encoded in RGB) and considering the resulting eigenvectors. The method for this is as follows:

Let  $\mathbf{S}$  be the image to be segmented,  $\mathbf{S} \in \text{RGB}$ .

Let  $\mathbf{\Lambda}$  be the covariance matrix of the distributions of R, G and B in  $\mathbf{S}$ .

Let  $\lambda_1, \lambda_2$ , and  $\lambda_3$  be the eigenvalues of  $\Lambda$  such that  $\lambda_1 \geq \lambda_2 \geq \lambda_3$ .

Let  $w_i = (w_{Ri} \ w_{Gi} \ w_{Bi})^t$  be the eigenvectors of  $\Lambda$  corresponding to  $\lambda_i$  for  $i = 1, 2$  and  $3$ .

The set of decorrelated colour features for a given image are obtained by:

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} w_{R1} & w_{G1} & w_{B1} \\ w_{R2} & w_{G2} & w_{B2} \\ w_{R3} & w_{G3} & w_{B3} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.1)$$

In Ohta's experiments,  $X_1$  has the largest variance (between 75% and 97% of the combined variance),  $X_2$  is the next best among features orthogonal to  $X_1$  (variance between 3% and 23% of the combined variance).  $X_3$  accounts for less than 4% of the combined variance, but its omission results in clear degradation in some areas of reconstructed images, as Ohta found in experiments using only the first two spectral features.

When Ohta examined the eigenvectors of  $\Lambda$  for each of the experimental images, it was found that each  $w_i$  is characterized by similar vectors across the entire set. Ohta found that  $w_1$  is approximately  $(1/3 \ 1/3 \ 1/3)^t$ ,  $w_2$  either  $(1/2 \ 0 \ -1/2)^t$  or  $(-1/2 \ 0 \ 1/2)^t$ , and  $w_3$  is  $(-1/4 \ 1/2 \ -1/4)^t$ .

This equates to three orthogonal colour features given by:

$$O_1 = \frac{R + G + B}{3} \quad (2.2)$$

$$O_2 = \frac{R - B}{2} \text{ or } \frac{B - R}{2} \quad (2.3)$$

$$O_3 = \frac{2G - R - B}{4} \quad (2.4)$$

or the following matrix transformation:

$$\begin{bmatrix} O_1 \\ O_2 \\ O_3 \end{bmatrix} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ \pm 1/2 & 0 & \mp 1/2 \\ -1/4 & 1/2 & -1/4 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.5)$$

In this work we shall assume usage of the first variant of  $O_2$ . The predominance of feature  $O_1$  shows that intensity is the most important feature even in colour image processing. To convert back to RGB, the following matrix may be used:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1 & -2/3 \\ 1 & 0 & 4/3 \\ 1 & -1 & -2/3 \end{bmatrix} \begin{bmatrix} O_1 \\ O_2 \\ O_3 \end{bmatrix} \quad (2.6)$$

The Ohta colourspace is advantageous in that the decorrelated components are useful for efficient encoding, and as Ohta demonstrated, well-suited to colour region segmentation. The transform from RGB is computationally efficient with implementation needing only a few multiplication and/or bit-shifting steps. Incidentally, a colourspace developed by Tan and Kittler using the same method produced almost identical results; the only difference being an inversion of the  $O_3$  vector of the Ohta colourspace.<sup>44</sup>

## 2.4 HSI-TYPE COLOURSPACES

There is a wealth of colourspaces based on the intuitive notion of specifying a colour by its hue, saturation and intensity. The concept of hue and saturation is most easily seen by rotating the RGB colour cube in Figure 2.1 and viewing the  $R = G = B$  axis as the  $z$ -axis through the page (Figure 2.2a). Hue can be specified as the angle around the centre where pure, fully saturated hues exist on the perimeter, and saturation is the distance from the centre. Overall intensity is the vertical axis (Figure 2.2b).

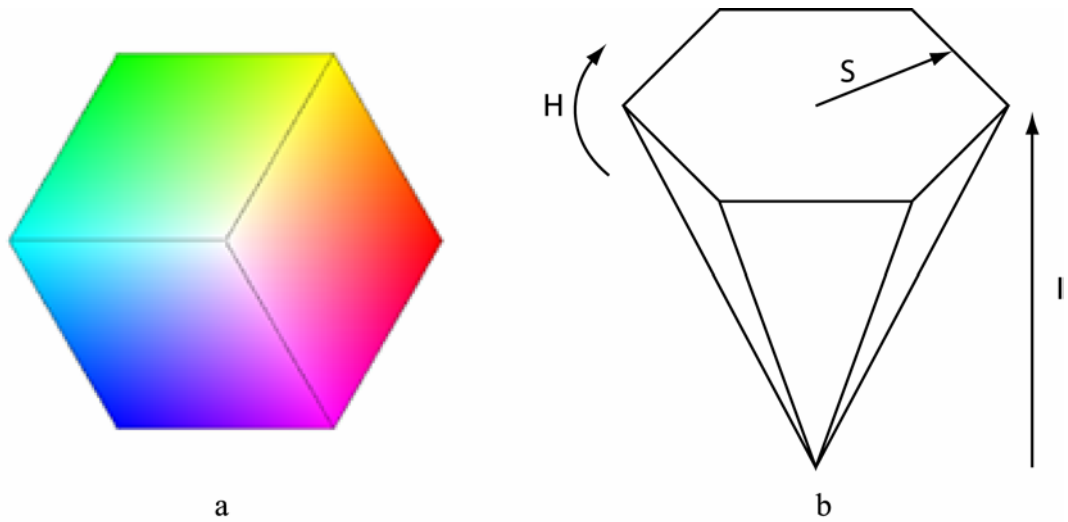


Figure 2.2. HSI colourspace hexcone.

The hue value ranges from  $0^\circ$  to  $360^\circ$ , and is calculated by the method of Gonzalez and Woods<sup>39</sup>:

$$H = \begin{cases} \cos^{-1} \frac{((R-G) + (R-B))/2}{(R-G)^2 + (R-B)(G-B)} & \text{if } G \geq B \\ 2\pi - \cos^{-1} \frac{((R-G) + (R-B))/2}{(R-G)^2 + (R-B)(G-B)} & \text{if } G < B \end{cases} \quad (2.7)$$

To obtain the saturation component, first the minimum value of the RGB triplet must be found:

$$S = 1 - 3 \frac{\min(R, G, B)}{R + G + B} \quad (2.8)$$

If  $S = 0$  then  $H$  is undefined. Intensity may be specified in the same fashion as the Ohta space, as the mean of the RGB triplet:

$$I = \frac{R + G + B}{3} \quad (2.9)$$

An alternative method of obtaining an HSI-type definition is given by Travis<sup>40</sup>, called HSV (Hue, Saturation, Value).

Minimum and maximum values from the RGB triplet must be found first. The Value component is set to the maximum, and the saturation is defined as the fractional difference between the maximum and minimum values:

$$S = \frac{(\max_{RGB} - \min_{RGB})}{\max_{RGB}} \quad (2.10)$$

$$V = \max_{RGB}$$

To calculate the hue, first fractional components of the RGB triplets are computed:

$$R_1 = \frac{\max_{RGB} - R}{\max_{RGB} - \min_{RGB}}$$

$$G_1 = \frac{\max_{RGB} - G}{\max_{RGB} - \min_{RGB}} \quad (2.11)$$

$$B_1 = \frac{\max_{RGB} - B}{\max_{RGB} - \min_{RGB}}$$

Again, if  $S = 0$  the resulting colour is monochrome and  $H$  is undefined, otherwise:

$$\begin{aligned} &\text{if } R = \max_{RGB} \text{ and } G = \min_{RGB} : \\ &\quad H = 5 + B_1 \\ &\text{if } R = \max_{RGB} \text{ and } G \neq \min_{RGB} : \\ &\quad H = 1 - G_1 \\ &\text{if } G = \max_{RGB} \text{ and } B = \min_{RGB} : \\ &\quad H = R_1 + 1 \\ &\text{if } G = \max_{RGB} \text{ and } B \neq \min_{RGB} : \\ &\quad H = 3 - B_1 \\ &\text{if } R = \max_{RGB} : \\ &\quad H = 3 + G_1 \\ &\text{otherwise:} \\ &\quad H = 5 - R_1 \end{aligned} \quad (2.12)$$

$S$  and  $V$  have ranges from 0 to 1 and  $H$ , when multiplied by 60, is expressed in degrees. It can be seen there are number of condition-testing steps to set the hue in this method, but this alleviates the need to use trigonometric functions or look-up

tables to generate values. Foley and van Dam list alternatives for HSI-type colourspaces.<sup>47</sup>

Although intuitive, and ideal as a means of colour specification for computer graphic artists, the non-linear transform from RGB causes a number of problems; hue is unstable for low intensity and saturation values, and spurious modes in the hue histogram arise due to the disjoint nature of the calculation.<sup>41</sup>

However, HSI-type colourspaces have been used with some success by a number of workers for colour segmentation applications.<sup>19,41,44,46</sup> It is also often used for histogram equalization of colour images, where only the  $I$  channel is considered.<sup>28,43,54</sup>

## 2.5 YC<sub>b</sub>C<sub>r</sub>

The YC<sub>b</sub>C<sub>r</sub> colourspace is an international standard for digital video broadcast and is also used in the JPEG and MPEG image formats.<sup>49</sup> The chrominance channels  $C_b$  and  $C_r$  and luminance channel  $Y$  are well decorrelated. In the JPEG standard and other DCT-based coding systems greater image compression is achieved by coarser quantization of the chrominance components than for that of the luminance.<sup>62</sup> Although a standard digital format, YC<sub>b</sub>C<sub>r</sub> does not specify any reference primaries; thus, no colour consistency can be guaranteed across display devices. To convert from RGB the following matrix is used:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & 0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.13)$$

A further standard was developed in 1988 which specifies system primaries and white point, but this is not yet recommended for use.<sup>50</sup>

## 2.6 CIE XYZ

As stated in Chapter 1, there is a combination of three visible primaries which can display a greater range of colours than any other, but it is not possible to reproduce the entire colour gamut visible to humans with these primaries.

In 1931, the CIE introduced the XYZ colourspace to address this problem<sup>51</sup>; the three primaries **X**, **Y** and **Z** can be used to match any visible colour using only positive weights. The use of only positive weights means that the primaries themselves are outside the visible range. Conversely, not all XYZ vectors are reproducible in other colourspaces and display devices.

The **X** and **Z** primaries specify chromaticity and the **Y** primary is intentionally defined to roughly match the human perception of luminance. The XYZ vector weights are typically normalized in the following manner:

If  $C$  is the colour to be reproduced, let  $C = XX + YY + ZZ$  where  $X$ ,  $Y$  and  $Z$  are all positive. Let new co-ordinates  $x$ ,  $y$  and  $z$  be defined such that  $x + y + z = 1$  (in this way only  $x$  and  $y$  need defining):

$$x = \frac{X}{X + Y + Z} \quad y = \frac{Y}{X + Y + Z} \quad (2.14)$$

Normalizing against  $X + Y + Z$  allows the chromaticity to be specified independently of the total luminous energy; colours are specified according to their chrominance  $xy$  and luminance  $Y$ . The CIE  $xy$  chromaticity diagram is shown in Figure 2.3. This colour space is highly non-linear, but has the advantage that all colours visible to the human eye are represented. The reverse relations to recover XYZ co-ordinates are:

$$X = \frac{x}{y} Y \quad Z = \frac{1 - x - y}{y} Y \quad Y = Y \quad (2.15)$$

To convert to XYZ from the EBU RGB space the following matrix is used<sup>28</sup>:



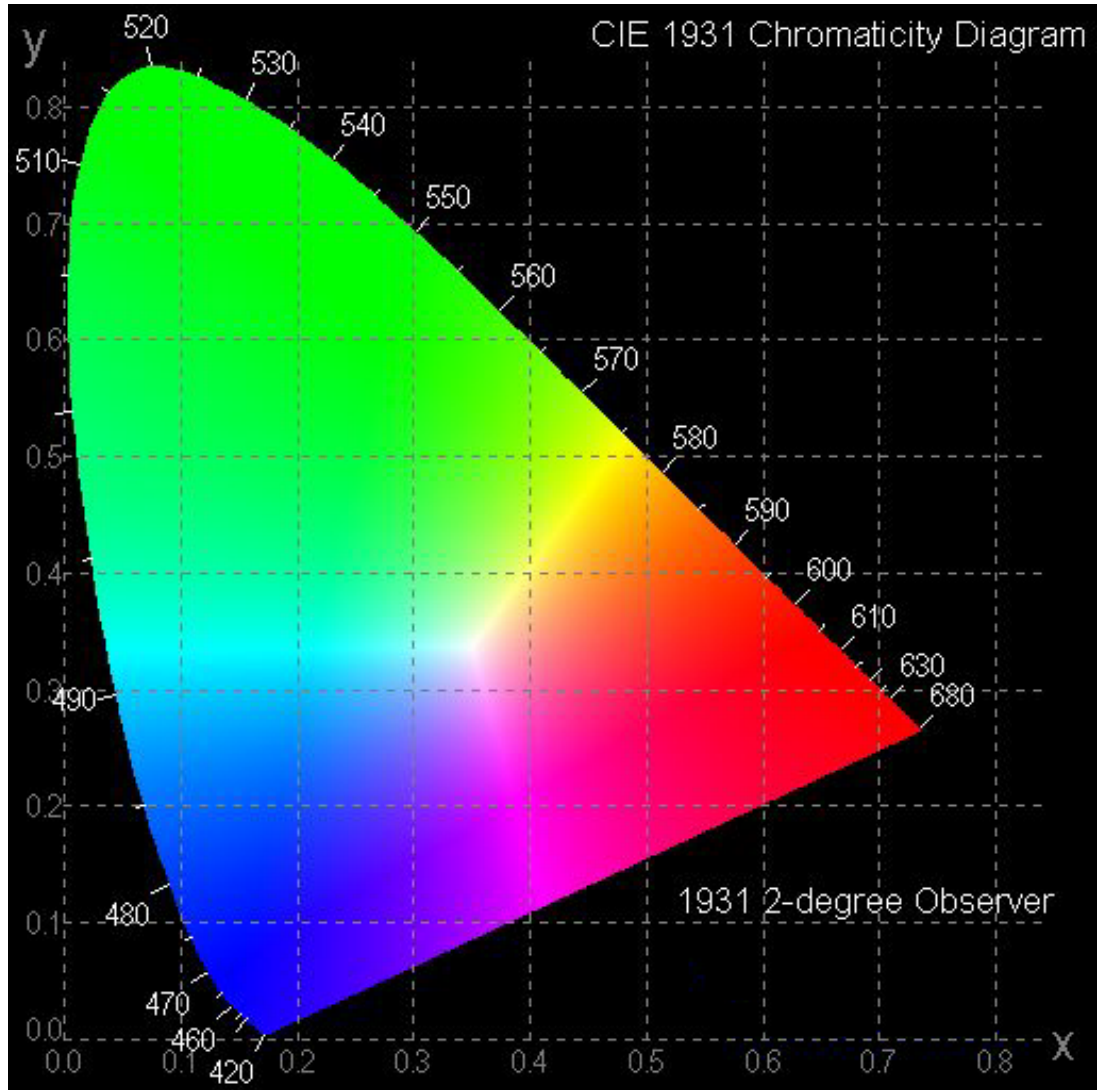


Figure 2.3. CIE (1931) Chromaticity diagram ([www.efg2.com](http://www.efg2.com)).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.430 & 0.342 & 0.178 \\ 0.222 & 0.707 & 0.071 \\ 0.020 & 0.130 & 0.939 \end{bmatrix} \begin{bmatrix} R_{EBU} \\ G_{EBU} \\ B_{EBU} \end{bmatrix} \quad (2.16)$$

And to obtain  $XYZ$  co-ordinates from  $RGB$ :

$$\begin{bmatrix} R_{EBU} \\ G_{EBU} \\ B_{EBU} \end{bmatrix} = \begin{bmatrix} 3.078 & -1.415 & -0.477 \\ -0.971 & 1.878 & 0.042 \\ 0.048 & -0.190 & 1.070 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.17)$$

Any  $XYZ$  co-ordinate that is out of the  $RGB$  gamut will produce a negative value. The  $XYZ$  colour space is the basis for a variety of other CIE colour spaces which attempt to achieve perceptual uniformity, in which equal vector distances in colour space are perceived as equal by a human observer. Other modifications have also been made to these colour spaces to permit more intuitive colour selection. A full discussion of the set of CIE colour spaces is beyond the scope of this thesis, but two of the most common examples used in colour image processing are described below.

## 2.7 CIE LUV

In 1976 as part of the evolving need to perceptually linearize the CIE colour space the LUV (sometimes denoted  $L^*u^*v^*$ ) colour space was developed, with primary application to emitted (additive) colour light sources, such as computer displays. The conversion from  $XYZ$  is non-linear but reversible, as follows:

$$L^* = 116f\left(\frac{Y}{Y_0}\right) - 16 \quad (2.18)$$

$$\begin{aligned} u^* &= 13L^*(u' - u'_0) \\ v^* &= 13L^*(v' - v'_0) \end{aligned} \quad (2.19)$$

where:

$$f(t) = \begin{cases} t^{\frac{1}{3}} & \text{if } t > 0.008856 \\ 7.787t^{\frac{1}{3}} + \frac{16}{116} & \text{if } t \leq 0.008856 \end{cases} \quad (2.20)$$

$$\begin{aligned} u' &= \frac{4X}{X + 15Y + 3Z} \\ v' &= \frac{9Y}{X + 15Y + 3Z} \end{aligned} \quad (2.21)$$

This colour space must have the white point  $X_0Y_0Z_0$  defined.

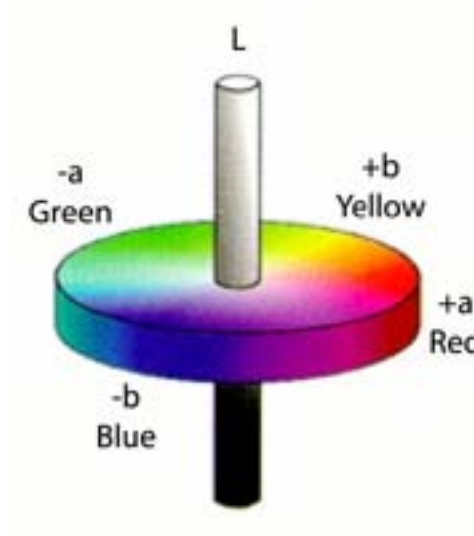


Figure 2.4. The LAB colour model ([www.cs.sfu.ca](http://www.cs.sfu.ca)).

## 2.8 CIE LAB

The LAB space (sometimes denoted  $L^*a^*b^*$ ) is another attempt at a perceptually uniform colourspace with the emphasis on accuracy for reflected light sources. As with LUV, the  $X_0Y_0Z_0$  vector represents reference white.

The orthogonal vectors  $a$  and  $b$  are opponent axes, with  $a$  representing the red–green axis and  $b$  the yellow–blue axis (Figure 2.4). The LAB components are:

$$L^* = 116f\left(\frac{Y}{Y_0}\right) - 16 \quad (2.22)$$

$$\begin{aligned} a^* &= 500 \left[ f\left(\frac{X}{X_0}\right) - f\left(\frac{Y}{Y_0}\right) \right] \\ b^* &= 200 \left[ f\left(\frac{Y}{Y_0}\right) - f\left(\frac{Z}{Z_0}\right) \right] \end{aligned} \quad (2.23)$$

where  $f(t)$  is defined in Eq. 2.20. Both the LAB and LUV colourspaces are frequently employed for a variety of machine vision applications; their popularity is

exemplified by the number of publications in which they have been used.<sup>20,55,56,57,58,59,60,61,63,65,66</sup>

## 2.9 CONCLUDING REMARKS

In this chapter, the concept of a colour space has been defined, along with descriptions of some important colour spaces relevant to colour machine vision. The properties of these colour spaces have been discussed with comparison of their features relative to one another, namely:

- Perceptual accuracy
- Interchannel correlation
- Computational load
- Device consistency
- Conversion issues
- Typical applications

The discussion of these factors highlights the suitability of a given type of colour space to a particular task.

# Chapter 3: The Colour Image Processing Environment

## 3.1 IMAGE PROCESSING ENVIRONMENT REQUIREMENTS

Necessarily, the environment in which algorithms are tested must fulfill a number of predefined requirements. First, these requirements are discussed and then compared with the functionality and specification of currently available software. Finally, details of the chosen implementation are given.

### 3.1.1 USE OF RESOURCES

The experimental framework should attempt to minimize use of system resources. It should not consume any more CPU resources than necessary to enable accurate evaluation of the algorithms under test. Excessive RAM usage leads to a slowdown of performance on multi-tasking systems and reduces available space for images. It should also be able to fit on common storage media to facilitate easy transfer between computers should an experiment necessitate comparisons to be made between different machines or should an experiment require the processing resources to be split between two or more CPUs.

### 3.1.2 MULTIPLE IMAGE DISPLAY

The system should be able to display multiple full colour images simultaneously to permit comparison of images between various processing stages.

### **3.1.3 STORAGE FORMAT COMPATIBILITY**

The image processing environment should be able to read and write image data in at least one common image format to permit viewing or comparison in other software packages.

### **3.1.4 IMAGE SIZE**

The framework should be able to work with images of arbitrary size and only be limited by available resources.

### **3.1.5 OPERABILITY**

The user interface of the system should not hamper algorithm development or testing. An interface that is easy to navigate and does not require complex interaction is therefore relatively important.

### **3.1.6 REAL-TIME APPLICABILITY**

Although not of prime importance, it should be possible to draw useful conclusions from performance in the test environment about an algorithm's suitability in a real-time and/or embedded environment. The test framework should therefore not include any optimizations that cannot be easily implemented on other platforms, or indeed, contain any peculiar performance constraints.

## **3.2 AVAILABLE SOFTWARE**

Whilst a real-time operating system would be desirable to obtain accurate performance measurement, the wide adoption of the Microsoft Windows operating system makes it a natural choice as an operating platform. Although there is an abundance of consumer-quality image processing software for Windows, there is a

paucity of semi-professional software which allows extra functionality to be added by the user. At the time of writing, only Photoshop by Adobe Systems Inc. and Matlab by The Mathworks Inc. allow the use of user-written functions. However, neither of these packages are a suitable solution: Photoshop's provision for additional functions ('plug-ins') must be written using a specialist API (Application Programming Interface) and the lack of GUI (Graphical User Interface) and comprehensive image manipulation functions in Matlab preclude them from serious consideration as colour image processing environments for the experiments undertaken in this thesis.

### **3.3 IMPLEMENTATION**

After the commercially available software was considered and discarded on the merits discussed above, the decision was taken to design and implement a bespoke colour image processing system.

The prevalence of the desktop PC and the Microsoft Windows operating system meant this combination was the obvious choice upon which to implement the image processing framework. For reasons of speed, flexibility and ability to perform low-level manipulation of data, the C programming language was used to develop the software.

To accommodate any number of images of differing sizes, a standard MDI (Multiple Document Interface) layout was employed; each image is contained within its own window (referred to as the child window) which may be moved or resized within the main application frame.<sup>152</sup>

At the heart of each Windows application is the 'message pump' which receives and translates messages generated by itself, the OS and other applications. For instance, each child window must be able to respond to options selected in the main application frame menu bar, and mouse clicks and movement within their respective child window areas. Once the basic framework has been constructed, i.e. the frame window creation and the standard code to handle messages, it is a fairly simple

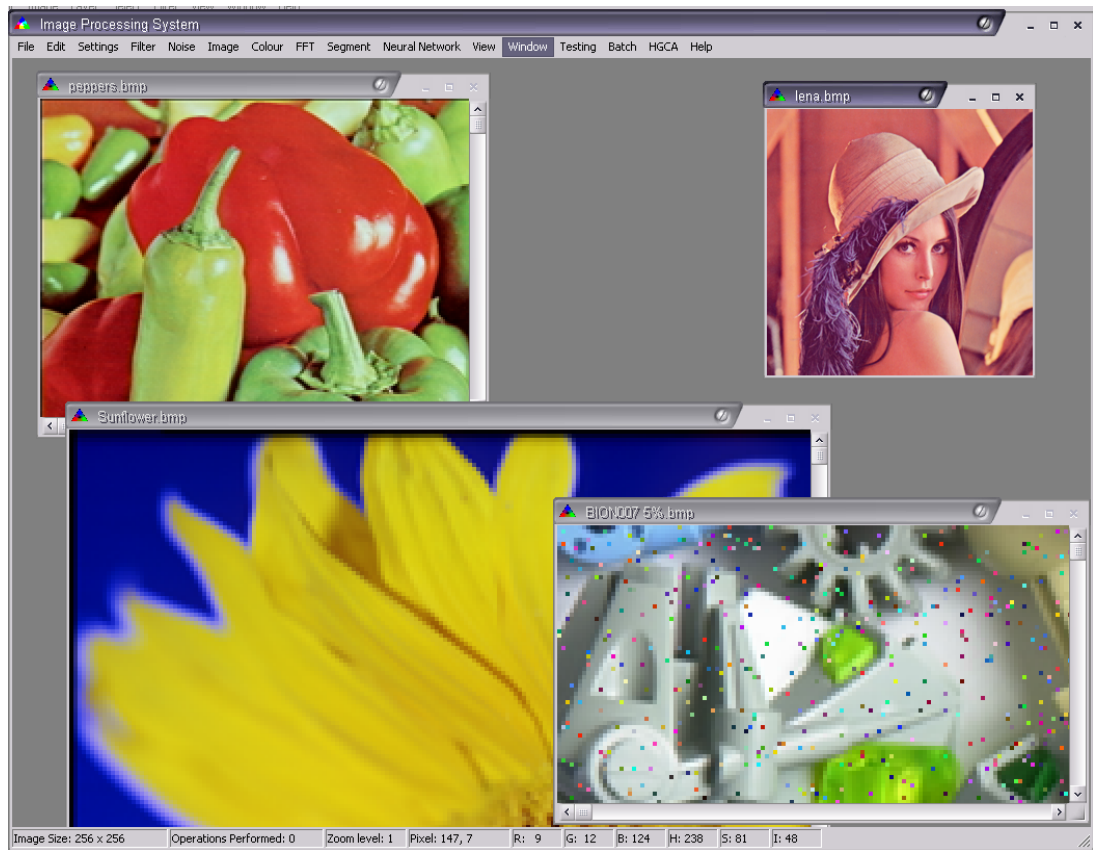
matter to quickly and easily add functionality by adding a menu item and handler code. For instance, the following C code shows simple example message handling code for a child window:

```
LRESULT CALLBACK ChildWindowProc (HWND hwnd, UINT iMsg, WPARAM wParam,
LPARAM lParam)
/* message parameters sent to child window handler include the unique
identifier handle of the window, the message type and two auxiliary
parameters to carry further message information */
{
    switch (iMsg)
    {
        case WM_COMMAND :
            // handle menu messages
            switch (wParam)
            {
                case MENU_ITEM_1 :
                    // do menu action
                case MENU_ITEM_2 :
                    // do other menu action, etc.
            }
        case WM_MOUSEMOVE :
            // handle mouse movements
        case WM_SIZE :
            // handle window resizing
        case WM_LBUTTONDOWN :
            // handle mouse left button double click (used for zooming in)
        case WM_RBUTTONDOWN :
            // handle mouse right button double click (used for zooming
            // out)
        case WM_MDIACTIVATE :
            // message received when focus changes to different child
            // window
        case WM_etc...
    }
}
```

Each child window contains an extended data area; for this application this is used to store a pointer to an image information structure, defined as:

```
typedef struct tagIMAGEDATA
{
    BOOL bNeedSave;           // check if image requires saving
    PSTR pImageTitle;         // window title of image
    int iImageVer;            // how many operations done
    LONG iXsize;              // horizontal image size
    LONG iYsize;              // vertical image size
    POINT pScrPos;            // current scroll offset
    RGBTRIPLE *RGBdata;       // pointer to RGB display data
    POINT pScrMax;            // current scroll limits
    int iZoom;                // current image zoom factor
} IMAGEDATA, *LPIMAGEDATA ;
```





**Figure 3.1. Example screenshot of image processing system.**

Each time an action is performed on a child window, a new child window is created with the result of the operation. This leaves the original image untouched and allows for easy comparison.

The image processing framework is designed to work with TrueColour (i.e. 24-bit) images; thus display adapter modes of both 24- and 32-bits per pixel are supported.

### **3.3.1 FUNCTIONALITY**

Two native file formats are supported for loading and saving images: raw RGB or greyscale byte data and the standard windows BMP file format (to ensure compatibility and to facilitate easy transfer to and from other Windows software packages). An example screen shot of IPS with some opened image files demonstrating the MDI layout is shown in Figure 3.1.

As well as providing a simple framework in which processing functions can be written, the image processing system ('IPS') contains a number of other useful features:

- Image zooming and cropping
- Copy and paste images via the windows clipboard direct from other applications
- 'Notepad'-style text window for data output
- Real-time display of pixel position and RGB values at mouse cursor position
- Graphic display of image RGB and HSI histograms
- Export image data and histogram as marker separated variables (suitable for Microsoft Excel or Matlab, for instance)

Each image manipulation function in IPS follows a basic format; an example function is shown below:

```
LPIMAGEDATA Subroutine(HWND hwnd, LPIMAGEDATA source)
{
    LPIMAGEDATA dest;
    int ix,iy,x,y;
    RGBTRIPLE *s, *d;

    ix=source->iXsize;
    iy=source->iYsize;

    // Create new image for child window of same size
    if ((dest=MakeImage(hwnd,ix,iy))==NULL) return 0;

    s=source->RGBdata; // Let s represent source image data
    d=dest->RGBdata; // Let d represent source image data

    // Typically will raster scan through image
    for(y=0;y<iy;y++) {
        for(x=0;x<ix;x++) {
            // Put your code here
            // For instance, swap red and blue channels
            d[x+y*ix].rgbtRed = s[x+y*ix].rgbtBlue;
            d[x+y*ix].rgbtBlue = s[x+y*ix].rgbtRed;
            d[x+y*ix].rgbtGreen = s[x+y*ix].rgbtGreen;
        }
    }
    // Clean up
    dest->bNeedSave=TRUE; // Image has changed, needs saving
    strcpy(dest->pImageTitle, source->pImageTitle); // Update title
    dest->iImageVer=source->iImageVer+1; // Increase op. count
    return dest; // return new image data
}
```

### **3.4 TEST IMAGES USED IN THIS THESIS**

To permit comparison with published work, the eight colour images from the Waterloo Repertoire Colorset from the University of Waterloo<sup>151</sup> have been employed in this thesis. The set contains both natural and ‘computer-art’ images depicting typical and pathological spatio-chromatic characteristics, providing comprehensive test data for machine vision researchers.

Where experimentation or further testing has required more images, an additional eight images selected to represent typical characteristics in natural scenes, industrial inspection and computer art have been chosen.

In experiments which require images to have similar statistical characteristics, two larger image sets of 32 images each have been used; one depicting plastic component pieces and the other showing confectionery pieces.

Both sets have features common to industrial inspection images; the confectionery image set contains simple objects: items of similar size and shape with a limited number of primary hues present. However, these properties are made more complex with the addition of specular and diffuse reflections, occlusions, shadows and damaged pieces.

The plastic component image set exhibits the same characteristics, but the items are of a more complex nature with a variety of shapes and sizes with a rich set of edge orientations and details.

Figure 3.2a-h shows the standard images from the Waterloo Repertoire Colorset, Figure 3.3a-h shows the additional eight images and Figures 3.4a-h and 3.5a-h show the first eight images of the plastic component and confectionery image sets.



Figure 3.2a. 'Clegg' image.

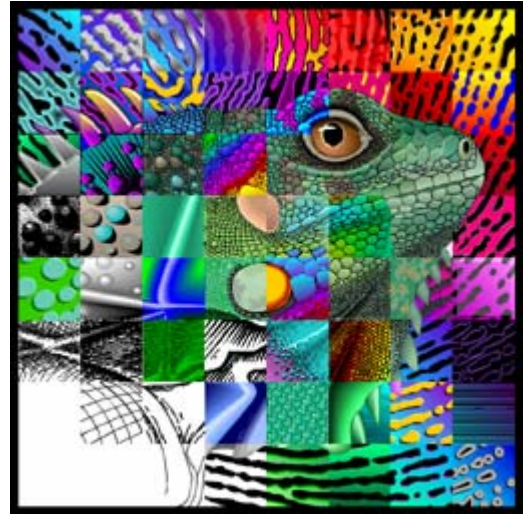


Figure 3.2b. 'Frymire' image.

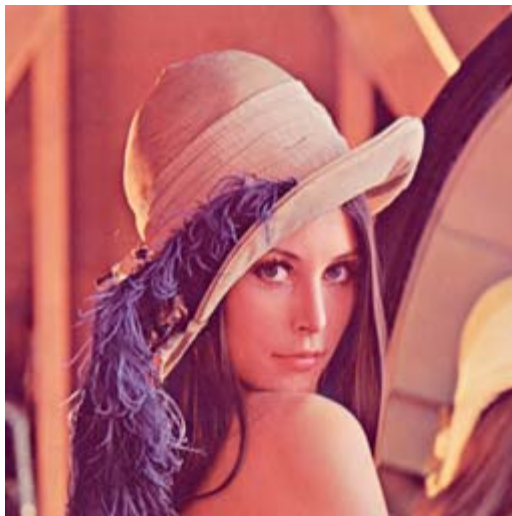


Figure 3.2c. 'Lena' image.



Figure 3.2d. 'Monarch' image.



**Figure 3.2e. ‘Peppers’ image.**



**Figure 3.2f. ‘Sail’ image.**



**Figure 3.2g. ‘Serrano’ image.**



**Figure 3.2h. ‘Tulips’ image.**





Figure 3.3a. 'Bion001' image.



Figure 3.3b. 'Hands' image.



Figure 3.3c. 'Italy03' image.



Figure 3.3d. 'Italy22' image.



**Figure 3.3e. ‘Jgdn2’ image.**



**Figure 3.3f. ‘MMs01’ image.**



**Figure 3.3g. ‘Phgdn1’ image.**



**Figure 3.3h. ‘Sunflower’ image.**



Figure 3.4a. 'Bion001' image.



Figure 3.4b. 'Bion002' image.



Figure 3.4c. 'Bion003' image.



Figure 3.4d. 'Bion004' image.





**Figure 3.4e. 'Bion005' image.**



**Figure 3.4f. 'Bion006' image.**



**Figure 3.4g. 'Bion007' image.**



**Figure 3.4h. 'Bion008' image.**



**Figure 3.5a. 'MMs01' image**



**Figure 3.5b. 'MMs02' image**



**Figure 3.5c. 'MMs03' image**



**Figure 3.5d. 'MMs04' image**



**Figure 3.5e. 'MMs05' image**



**Figure 3.5f. 'MMs06' image**



**Figure 3.5g. 'MMs07' image**



**Figure 3.5h. 'MMs08' image**

### **3.5 CONCLUDING REMARKS**

This chapter has outlined the requirements and basic implementation details of the image processing system used to execute the algorithms described in this thesis, and the images used and the rationale for their selection.

The program framework has proved to be ideally suited to the task and has been used successfully for modifying images, making quantitative and timing measurements and outputting data in formats suitable for importing into other software packages.

The program code containing the program framework and functions is included in Appendix B.

# Chapter 4: Local Window Methods of Noise Removal

## 4.1 DEFINITION OF THE LOCAL WINDOW

The local window is defined as all the pixels within a specified distance of a given pixel  $f(x,y)$ . As the vast majority of image geometry is Cartesian, it is natural to consider the local window as bounded by a square or rectangle. Although two-dimensional, for ease of notation the local window structure is generally considered as a 1-D array with the  $f(x,y)$  origin denoted  $W(0)$ ; this is given either as the top-left pixel with the index monotonically increasing in raster fashion (Figure 4.1a) or as the central pixel with the index increasing in a counter-clockwise spiral (Figure 4.1b). In this way each  $W(n)$  corresponds to an image location  $f(x+n_i, y+n_j)$ . Clearly, for pixels on or near the perimeter of the image, some values of  $n$  will indicate pixels which do not exist in the image. For these pixels the support of the window  $W$  should be curtailed or processing of the window abandoned entirely. Changing the window size dynamically clearly leads to different results for pixels near the edge than one would expect for the standard operator for the rest of the image. For this reason, processing of edge pixels is ignored in this thesis; for instance, if work is conducted with operators of maximum width  $2R+1$ , only pixels within distance  $R$  of the perimeter of the image are processed and subjected to scrutiny.

Local window noise removal methods replace or alter certain pixels within the window after considering the local image statistics; as most filtering techniques involve replacing the central pixel, local window sizes are generally of odd order (commonly from  $3 \times 3$  up to approximately  $21 \times 21$ ).



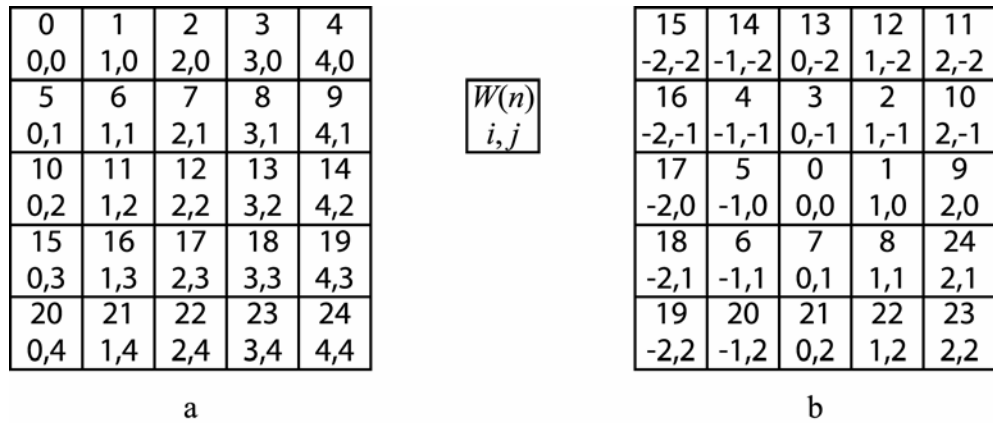


Figure 4.1. Two common forms of image sub-window notation.

For this same reason, the second window number ordering convention (Figure 4.1b) is employed in this thesis, as the index and relative location of the central pixel does not change with the operator size.

Rectangular or square local windows can be specified quite conveniently as shown above, but their support is anisotropic due to their geometry; more distance is covered by the diagonal than the horizontal or vertical. This may be useful for some applications; for instance, modern graphics cards include anisotropic filters with greater horizontal support than vertical for increased image quality or to produce effects such as motion blur.<sup>71</sup> However, anisotropic support is not desirable when measuring the efficacy of processing methods requiring accurate metrics; the effect of any technique should produce equal results in all directions, i.e. be invariant to image rotation.<sup>17,69</sup>

It is easy enough to specify pseudo-circular operators by only including sub-window elements which fall within a certain radius, although checking for this introduces an extra algorithmic delay which is unwelcome for time-critical applications. To this end octagonal windows can be employed which provide a useful trade-off between isotropic action and computational speed.<sup>70</sup>

## 4.2 CONVOLUTION OPERATORS

Low-pass filtering an image is a common method of removing or attenuating noise. Such filtering operations require that the image be transformed into the frequency domain, multiplied by the filter transfer function and then transformed back into the spatial domain.

However, it is well-known that a multiplication in the frequency domain is equivalent to a convolution in the time domain, and vice versa:<sup>1</sup>

$$\begin{aligned} S(\Omega)T(\Omega) &\Leftrightarrow s(x)*t(x) \\ S(\Omega)*T(\Omega) &\Leftrightarrow s(x)t(x) \end{aligned} \quad (4.1)$$

where  $S$  and  $T$  are the forward Fourier transforms of the spatial variables  $s$  and  $t$  respectively. In this way much of the computational overhead of transforming to and from the frequency domain can be eliminated by calculating the appropriate convolution mask from the filter frequency response for all but the largest masks.

The convolution of two functions  $f(x)$  and  $g(x)$  is defined as:

$$f(x)*g(x) = \int_{-\infty}^{\infty} f(u)g(x-u)du \quad (4.2)$$

For application to 2-D digital images the discrete form is given by:

$$F(x, y) = f(x, y) * g(x, y) = \sum_i \sum_j f(i, j)g(x-i, y-j) \quad (4.3)$$

To keep with the windowing notation and for a more intuitive understanding of the process of convolution, it is necessary to pre-invert the spatial convolution mask  $g$ :

$$h(x, y) = g(-x, -y) \quad (4.4)$$

$$F(x, y) = \sum_i \sum_j f(x+i, y+j)h(i, j) \quad (4.5)$$

In the literature, convolution masks are described in this pre-inverted form. It is common for the sum of the elements in the convolution mask to be unity so that there is no overall increase or decrease in image energy.

### 4.2.1 THE MEAN FILTER

The mean filter is the most basic of convolution masks to construct: each mask element contributes  $1/N^2$  to the output pixel for a mask of size  $N \times N$ . For instance, the mask for a  $3 \times 3$  operator is given by:

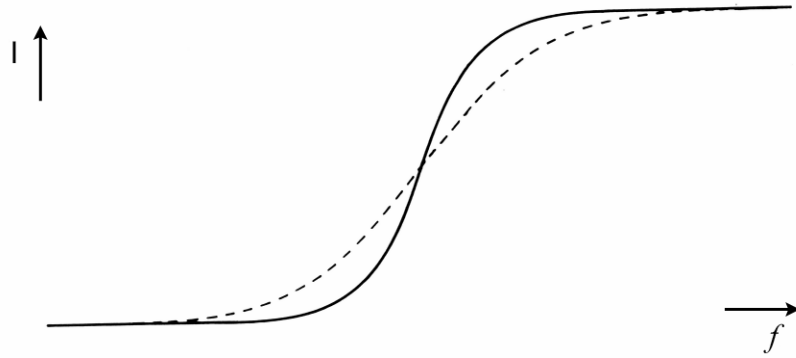
$$M_{mean} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Using the windowing convention in Fig. 4.1b, the output corresponds to the central pixel,  $W(0)$ .

As the output of the mean filter contains an equal proportion of all pixels in the window, the consequence is to distribute the noise present throughout the neighbourhood, effectively suppressing it. Indeed, it is well known that the mean filter is the minimum mean-squared error estimator for an unvarying signal contaminated with zero-mean Gaussian noise.<sup>72, 74</sup>

Unfortunately severe blurring occurs as adjacent pixels are mixed to produce the new central pixel value. This effect can be seen more clearly by considering Figure 4.2, which shows the intensity profile of a typical step edge after application of the mean filter: image detail is merely ‘averaged’ in. It naturally leads from this that the mean filter will perform poorly at removing impulsive (long-tailed) noise which exhibits similar high spatial-frequency characteristics to edge detail. The result is that the error introduced by the impulse will be spread out rather than eliminated, as previously noise-free pixels are averaged with the noisy ones.





**Figure 4.2. Blurring effect of the mean filter: the solid line is the original step-edge intensity profile.**

Selecting a convolution mask with a Gaussian or binomial profile ameliorates the difficulty somewhat (as well as approximating a circular symmetric operator) but is still far short of optimal:

$$M_{binomial} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Being a linear operation, the mean for colour image data can be obtained by simple component-wise application of the filter, for example:

$$\begin{aligned} m_r &= \frac{1}{N} \sum_{i=0}^N x_{r_i} \\ m_g &= \frac{1}{N} \sum_{i=0}^N x_{g_i} \\ m_b &= \frac{1}{N} \sum_{i=0}^N x_{b_i} \end{aligned} \tag{4.6}$$

such that  $\mathbf{m}$  is the mean vector of all  $\mathbf{x}_i$ .

### 4.3 THE MEDIAN FILTER

From the discussion above it is apparent that the mean filter is somewhat ineffective against impulsive noise; the median filter provides a simple and effective solution to

the problem. First brought to popular attention by Tukey,<sup>75</sup> the median filter belongs to the class of non-linear filters and operates over the local window in a manner similar to the mean filter, where the central pixel value is replaced by the median of the data set.

The advantage of the median filter is that its impulse response is zero, whilst step and ramp inputs pass through unchanged; this suppresses impulse noise extremely effectively without destroying image detail excessively; images tend to be slightly ‘softened’ which is considered more acceptable than the blurring which occurs with the mean filter.

For a continuous signal, the median  $x_{med}$  can be defined as the value which satisfies the following equation:

$$\int_{x_{min}}^{x_{med}} x \, dx - \int_{x_{med}}^{x_{max}} x \, dx = 0 \quad (4.7)$$

where  $x_{min}$  and  $x_{max}$  represent the range of values over which  $x$  lies.

In greyscale image processing the median of a local set of  $N$  pixels can be calculated simply by sorting the pixels according to intensity and selecting the pixel in the middle of the range, at position  $N/2$ . This is a unique pixel value if one constrains the median to be one of the original pixels in the distribution and ensures that  $N$  is odd.

As the median generally lies close to the mean except for certain pathological cases (ordering of the averages is discussed further in Chapter 6; see also Davies (1988)<sup>134</sup> for a further discussion), median filtering also has some success in curbing heavy Gaussian noise but is suboptimal in comparison to the mean;<sup>80</sup> with its excellent impulse rejection capability its relative popularity in suppressing noise when the noise and data characteristics are unknown can be understood.

However, due to its nonlinear operation, applying the median filter to colour images cannot be realized by simply extending its operation to each channel individually as with linear operators such as the mean filter. Such a method fails to take into account the interdependency of image components. Consider Figure 4.3a which

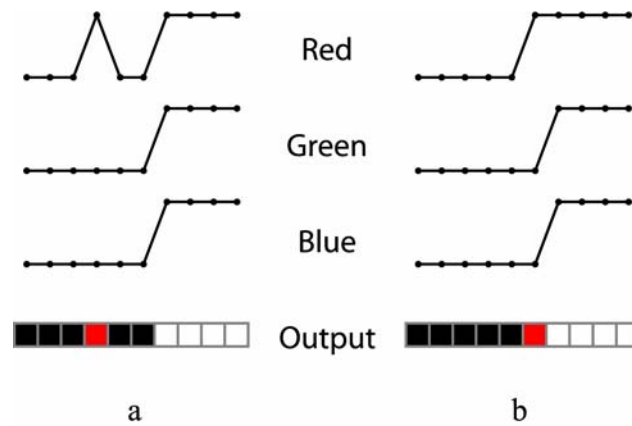


Figure 4.3. Application of 5-element median filter to colour pixel sequence.

shows a step edge from a small horizontal slice of a colour image, contaminated by a nearby impulse in the red channel. When this is filtered in a component-wise manner by a 5-element median filter (shown in Figure 4.3b) it can be seen that although the noise point is removed, it has caused the edge to move by one pixel; this is known as edge jitter. In this case the component-wise application of the scalar median filter has not truly succeeded in eliminating the noisy pixel; it has merely moved it. Also undesirable is that the output vector may not consist of one of the original input vectors for more complex examples. To overcome this problem, a vector median method utilizing the inter-channel correlation has been developed.

## 4.4 THE VECTOR MEDIAN FILTER

When designing a well-behaved (i.e. non-oscillatory) filter it is necessary to consider certain criteria. Before proceeding with definitions for the vector median filter, let us regard these criteria:

- The existence of root signals, i.e. those which are invariant to the application of the filter, must be assured to ensure the filter's stability. Root signals are important in the analysis of a filter's behaviour, as essentially they define the filter passband.

- In the 1D case, a vector filter should produce identical output to that of the scalar version, i.e. select the same output value.
- As with the scalar filter, the output of a vector filter should be one of the original input vectors (thus ameliorating the problem of edge jitter, or colour-bleeding).

With these properties defined, the vector median may be identified either using robust ordering statistics or as a maximum likelihood estimator.

#### **4.4.1 DEFINITION 1: DERIVING THE VECTOR MEDIAN USING RANK ORDER STATISTICS**

As there is no natural concept of ordering vector-valued quantities, some method must be employed by which one may rank order the values to obtain the ‘median’ for a multivariate data set. To extend scalar processing to vector data, four distinct ordering techniques are commonly used: conditional ordering (C-ordering), marginal ordering (M-ordering), partial ordering (P-ordering) and reduced ordering (R-ordering).<sup>76</sup>

In C-ordering, the data are ranked according to one component only; this may conceivably be employed for images encoded in certain colourspaces whereby one component is more statistically significant (see Chapter 2) but cannot be considered suitable for general colour image processing use as it fails to take into account inter-component correlation. However, at least one filtering method has been developed using C-ordering in the HSV colourspace.<sup>77</sup> In M-ordering, ranking is performed for each component independently; obtaining the median by component-wise processing is unsuitable for the reasons described in Section 4.3. In P-ordering, the convex hull of the data set is found and all samples on the boundary are assigned rank 1 and then discarded. A new convex hull is found and samples assigned rank 2 before being discarded. This continues until all the data have been classified. P-ordering does not distinguish between samples which have been assigned the same rank and calculating the convex hull for  $M$ -dimensional distributions is complicated

and computationally expensive for  $M \geq 3$ . In R-ordering each vector is assigned a scalar value according to a pre-specified dissimilarity or distance metric  $D$ :

$$d_j = \sum_{i=1}^N D(\mathbf{x}_i, \mathbf{x}_j) \quad (4.8)$$

The vector which has the lowest distance  $d$  is assigned rank 1, the second lowest rank 2, and so on, such that:

$$\mathbf{x}_1 \leq \mathbf{x}_2 \leq \dots \leq \mathbf{x}_N$$

corresponds to:

$$d_1 \leq d_2 \leq \dots \leq d_N$$

The R-ordering approach has the advantages that each sample generally has unique rank and the distance criterion can be chosen according to application. It is probable that for these reasons a great many publications utilize a form of the R-ordering rank statistic in colour image processing.<sup>28,72,73,74,77,78,79</sup>

Simple analysis of the scalar median filter shows that for a signal  $x$  of length  $N$ , the median filter output  $x_{med}$  satisfies the following equation:

$$\sum_{i=1}^N |x_{med} - x_i| \leq \sum_{i=1}^N |x_j - x_i| \quad \forall x_j \quad (4.9)$$

It follows that a definition of the VMF may be obtained by modifying Eq. 4.9 to employ an  $L_p$  norm metric as the R-ordering measure:

$$\mathbf{x}_{med} = \min \sum_{i=1}^N \|\mathbf{x}_i - \mathbf{x}_j\|_p \quad \forall \mathbf{x}_j \quad (\mathbf{x}_{med} \equiv \mathbf{x}_1) \quad (4.10)$$

where the  $L_p$  norm is denoted by  $\|\cdot\|_p$  which for  $M$ -dimensional vectors is defined as:

$$L_p(u, v) = \left( \sum_{i=1}^M |u_i - v_i|^p \right)^{\frac{1}{p}} \quad (4.11)$$

Popular  $L_p$  norms are the  $L_1$ ,  $L_2$  and  $L_\infty$  metrics (the city-block, Euclidean and chessboard distances, respectively). Using an  $L_p$  metric as the R-ordering statistic means that vectors which diverge greatly from the sample population are found at the higher indexes; outlier rejection is inherent in its performance. The  $L_1$  norm approximates the component-wise application of the scalar median filter (Eq. 4.9) and the  $L_2$  norm, selecting the vector closest to the centre of the distribution, is more an approximation of the mean filter.<sup>81</sup> The  $L_1$  norm can expect to perform well in rejecting impulse noise, but less favourably in comparison to the  $L_2$  norm in a Gaussian environment (at the expensive of greater softening).<sup>81,82,83</sup>

#### 4.4.2 DEFINITION 2: DERIVING THE VECTOR MEDIAN AS A MAXIMUM LIKELIHOOD ESTIMATOR

Consider a  $M$ -dimensional sample set in which the noise in the components is dependent and symmetrically distributed biexponentially:<sup>83</sup>

$$f(\mathbf{x}_i) = \gamma e^{-\alpha \|\mathbf{x}_i - \boldsymbol{\beta}\|_2} \quad (4.12)$$

where  $\alpha$  and  $\gamma$  are scaling factors and  $\boldsymbol{\beta}$  is the location parameter of the distribution.

The maximum likelihood estimator  $\hat{\boldsymbol{\beta}}$  for  $\boldsymbol{\beta}$  maximizes the function:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^N \gamma e^{-\alpha \|\mathbf{x}_i - \boldsymbol{\beta}\|_2} \quad (4.13)$$

By taking the logarithm of Eq. 4.13, maximizing  $L(\boldsymbol{\beta})$  reduces to minimizing the equation:<sup>74</sup>

$$\sum_{i=1}^N \|\mathbf{x}_i - \boldsymbol{\beta}\|_2 \quad (4.14)$$

Astola et al note that there is no closed solution to the above equation unless  $\hat{\mathbf{b}}$  is constrained to be one of the set  $\mathbf{x}$ , in which case one arrives at equation 4.10 using the  $L_2$  norm.<sup>74</sup>

Similarly, they show that the  $L_1$  norm may be derived by considering a sample distribution in which the noise is independent in the  $M$  components:

$$f(\mathbf{x}) = \prod_{i=1}^M \frac{\alpha_i}{2} e^{-\alpha_i |x_i - \beta_i|} \quad (4.15)$$

where  $\alpha_i$  are the scaling factors. The expression to be minimized then becomes:

$$\sum_{i=1}^N \sum_{d=1}^M \alpha_d |x_{id} - \beta_d| \quad (4.16)$$

where  $x_{id}$  and  $\beta_d$  are the  $d$ 'th components of  $\mathbf{x}_i$  and  $\mathbf{b}$  respectively.

With the constraint that the maximum likelihood estimate  $\hat{\mathbf{b}}$  be one of  $\mathbf{x}_i$  and that  $\alpha_1 = \alpha_2 = \dots = \alpha_M$  one arrives at the definition of the  $L_1$  norm VMF:

$$\mathbf{x}_{med} = \min \sum_{i=1}^N |\mathbf{x}_i - \mathbf{x}_j| \quad \forall \mathbf{x}_j \quad (4.17)$$

With these definitions of the VMF it is possible that  $\mathbf{x}_{med}$  is not unique. In this case additional consideration must be given to candidate vector selection. In this work pixels are calculated in window number order (shown in Figure 4.1b) and thus of those  $\mathbf{x}_i$  which exhibit the lowest distance the one nearest the central pixel is favoured.

## 4.5 OTHER VECTOR FILTERS

The class of filters based on order statistics is rich, and although the VMF as discussed is a powerful noise removal tool that may be applied when the source and noise characteristics are not known *a priori*, it is likely that filtering according to

other priorities or properties of colour images is desirable. Here follow descriptions of a number of well-known non-linear and hybrid vector filters and their rationales.

#### 4.5.1 THE $\alpha$ -TRIMMED VECTOR MEDIAN FILTER ( $\alpha$ VMF)

As already noted in Sections 4.2.1 and 4.4.1 the excellent performance of the VMF against long-tailed or impulsive noise cannot approach that of the Arithmetic (linear) Mean Filter (AMF) for short-tailed (i.e. Gaussian) noise. This filter attempts to combine the effectiveness of the AMF for Gaussian contamination with the edge-preserving and impulse rejection properties of the VMF by passing the  $n(1-2\alpha)$  vectors closest to the vector median to an averaging filter block (Plataniotis & Venetsanopoulos, Chapter 10, *The Image Processing Handbook*).<sup>28</sup>

Assuming the input samples have already been sorted by  $L_p$  norm ( $\mathbf{x}_1 \leq \mathbf{x}_2 \leq \dots \leq \mathbf{x}_N$ ), the  $\alpha$ -trimmed VMF can be described thus:

$$\mathbf{x}_{\alpha VMF} = \sum_{i=1}^{n(1-2\alpha)} \frac{1}{n(1-2\alpha)} \mathbf{x}_i \quad (4.18)$$

where  $\alpha \leq 0.5$  and  $n\alpha \in \mathbb{N}$ .<sup>84,85</sup> The efficacy of this filter against the other filters described is discussed in Section 4.7.

#### 4.5.2 THE BASIC VECTOR DIRECTIONAL FILTER (BVDF)

The VMF is concerned with the differences in vector lengths, i.e. the aggregate vector intensity, and although it is well known that humans are more sensitive to errors in the luminance channel than the chrominance channel, it should not be forgotten that colour integrity is important when considering colour images, and thus the vector directional filter was developed in an attempt to eliminate those vectors inside the processing window which exhibit atypical directions, i.e. those which are of a clearly different *colour* to the surrounding pixels.



The basic vector directional filter (BVDF) uses the same R-ordering statistic, but instead of selecting the target pixel as that which minimizes the sum of the vector lengths, it picks the vector which minimizes the sum of the angles with the other vectors,  $a_i$ .

$$a_i = \sum_{j=1}^n A(\mathbf{x}_i, \mathbf{x}_j) \quad (4.19)$$

where

$$A(\mathbf{x}_i, \mathbf{x}_j) = \cos^{-1} \left( \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|} \right)$$

It is clear that this filter is only of use when preserving only the chromaticity of the image is important; as a preprocessing step before segmentation, or when it is known *a priori* that the luminance information is uncorrupted, for instance. Otherwise a subsequent magnitude processing operation should be implemented, as described in the more generalized filter structure described below.

### 4.5.3 THE GENERALIZED VECTOR DIRECTION FILTER (GVDF)

The GVDF combines the directional or chromatic filtering property of the BVDF with the length or intensity filtering property of the VMF by passing the set of  $r$  smallest sum-of-angles vectors obtained from the BVDF to an intensity processing structure. It should be noted that any type of magnitude processing may be implemented; Trahanias and Venetsanopoulos employ  $\alpha$ -trimmed mean and multistage max/median as secondary filters.<sup>86</sup>

Selection of  $r$  may be fixed or determined adaptively. The extreme cases  $r = 1$  corresponds to the output of BVDF only, and  $r = N$  corresponds to no directional filtering at all. In the experiment conducted in Section 4.7 the standard  $L_2$  norm metric is used for the magnitude processing, with  $r$  fixed at  $N/2 + 1$  as suggested by Trahanias et al.<sup>87</sup>

#### 4.5.4 THE DIRECTIONAL DISTANCE FILTER (DDF)

The DDF adopts an approach similar to the GVDF with regard to combined filtering of intensity and chromaticity, but rather than using a two-tier filter structure a new distance criterion is defined for the R-ordered VMF.<sup>88</sup>

$$\mathbf{x}_{out} = \min \left( \sum_{i=1}^n \sum_{j=1}^n \cos^{-1} \left( \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{|\mathbf{x}_i| |\mathbf{x}_j|} \right) \right) \left( \sum_{i=1}^n \sum_{j=1}^n \|\mathbf{x}_i - \mathbf{x}_j\| \right) \quad (4.20)$$

This is a combination of the BVDF and VMF; the first part of Eq. 4.20 accounts for differences in vector angle and the latter in terms of magnitude.

#### 4.5.5 THE DIRECTIONAL MAGNITUDE VECTOR FILTER (DMVF)

Another filter which modifies the distance criterion of Eq. 4.8 is the DMVF.<sup>82</sup> Based on the ‘content model’ proposed by Ekman,<sup>89</sup> a distance criterion is derived which measures the similarity between vector pairs. This is expressed as the ratio between the factors that they share to that of their combined total:

$$D(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sqrt{|\mathbf{x}_i|^2 + |\mathbf{x}_j|^2 - 2|\mathbf{x}_i||\mathbf{x}_j|\cos\theta}}{\sqrt{|\mathbf{x}_i|^2 + |\mathbf{x}_j|^2 + 2|\mathbf{x}_i||\mathbf{x}_j|\cos\theta}} \quad (4.21)$$

It can be seen that Eq. 4.21 considers both vector angle and length differences as with the DDF and GVDF. Plataniotis et al give results which shows the DMVF outperforms the other vector filters described so far (VMF, BVDF, GVDF and DDF) for both Gaussian and impulsive noise.<sup>82</sup>

#### 4.5.6 ADAPTIVE VECTOR FILTERS

All the filters discussed so far have been derived from particular viewpoint or constraints, or developed with specific applications in mind, and thus may perform

poorly in conditions other than those in which they were originally designed to operate. The construction of an adaptive filter structure which performs well for different image classes and noise models is therefore highly desirable. A general form of adaptive filter put forward by Plataniotis et al<sup>90</sup> is given as the nonlinear weighted average of the input samples:

$$\hat{\mathbf{y}} = \sum_{i=1}^N w'_i \mathbf{x}_i = \frac{\sum_{i=1}^N w_i \mathbf{x}_i}{\sum_{i=1}^N w_i} \quad (4.22)$$

where  $\hat{\mathbf{y}}$  is the filter output,  $w'_i \geq 0$  and  $\sum_{i=1}^N w'_i = 1$ .

The weights are determined adaptively according to a distance criterion in the same way that the other vector filters rank their input vectors, and a membership function which, based on the distance criterion, ensures that the input vector with the minimum distance is assigned the maximum weight  $w'_i$ .

From this premise is it possible to construct a plethora of filters utilizing different membership functions and distance criteria. Arguably, one of the most successful of this type of filter is the Adaptive Nearest Neighbour Multichannel Filter (ANNMF), in which weights are assigned according to the nearest neighbour rule:

$$w_i = \frac{d_{\max} - d_i}{d_{\max} - d_{\min}} \quad (4.23)$$

where the  $d_i$  are calculated as:

$$d_i = \sum_{j=1}^N 1 - \left( \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|} \right) \left( 1 - \frac{\|\mathbf{x}_i\| - \|\mathbf{x}_j\|}{\max \|\mathbf{x}_i\| \|\mathbf{x}_j\|} \right) \quad (4.23)$$

In tests against other vector filters, adaptive vector filters (and in particular the ANNMF) produce consistently lower error scores than the traditional vector filters.<sup>28</sup>

## 4.6 NOISE MEASUREMENT METRICS

To evaluate the efficacy of a given filter, objective error measures are required. Classic error metrics used in greyscale image processing are the mean squared error (MSE), root mean squared error (RMSE) and signal-to-noise ratio (SNR):

$$MSE = \frac{\sum_{x=1}^N \sum_{y=1}^M (f(x, y) - f'(x, y))^2}{NM} \quad (4.24)$$

$$RMSE = \frac{\sum_{x=1}^N \sum_{y=1}^M \sqrt{(f(x, y) - f'(x, y))^2}}{NM} \quad (4.25)$$

$$SNR = \frac{\sum_{x=1}^N \sum_{y=1}^M f'(x, y)^2}{\sum_{x=1}^N \sum_{y=1}^M (f(x, y) - f'(x, y))^2} \quad (4.26)$$

where  $N$  and  $M$  are the image width and height, and  $f(x, y)$  and  $f'(x, y)$  are the original and post-filtered noisy image pixels, respectively.

In colour image processing, two metrics are generally used: the normalized mean square error (NMSE), and the mean chromaticity error (MCRE). The normalized mean square error for vector images is defined as:<sup>86</sup>

$$NMSE = \frac{\sum_{x=1}^N \sum_{y=1}^M \|\mathbf{f}(x, y) - \mathbf{f}'(x, y)\|^2}{\sum_{x=1}^N \sum_{y=1}^M \|\mathbf{f}(x, y)\|^2} \quad (4.27)$$

The mean chromaticity error is generally calculated as the distance between the intersection points of two vectors on the Maxwell triangle (plane of constant intensity), as shown in Figure 4.4. The Maxwell triangle is defined as the plane  $x + y + z - 1 = 0$ , or in the RGB cube  $R + G + B - 255 = 0$  for images utilizing 8-bits per channel.

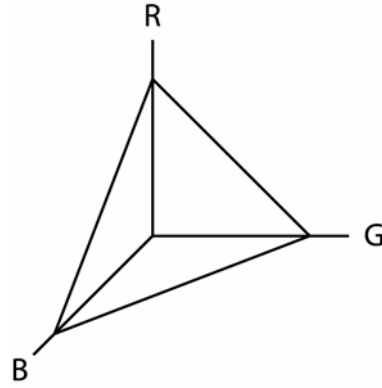


Figure 4.4. The Maxwell triangle.

The MCRE is thus defined as:<sup>86</sup>

$$MCRE_1 = \frac{\sum_{x=1}^N \sum_{y=1}^M \sqrt{(r_{xy} - \bar{r}_{xy})^2 + (g_{xy} - \bar{g}_{xy})^2 + (b_{xy} - \bar{b}_{xy})^2}}{NM} \quad (4.28)$$

Where  $r$ ,  $g$  and  $b$  are the normalized RGB values of a given pixel  $i$ :

$$r_i = \frac{R_i}{R_i + G_i + B_i} \quad g_i = \frac{G_i}{R_i + G_i + B_i} \quad b_i = \frac{B_i}{R_i + G_i + B_i}$$

Some workers utilize a slightly different method of calculating the MCRE which involves measuring the Euclidean distance between two points on the unit sphere ( $x^2 + y^2 + z^2 - 1 = 0$ ), rather than the unit plane:<sup>91</sup>

$$MCRE_2 = \frac{\sum_{x=0}^N \sum_{y=0}^M \left\| \frac{\bar{\mathbf{f}}(x, y)}{\|\bar{\mathbf{f}}(x, y)\|} - \frac{\mathbf{f}(x, y)}{\|\mathbf{f}(x, y)\|} \right\|^2}{NM} \quad (4.29)$$

If the definition of the MCRE in Eq. 4.29 is used in preference to Eq. 4.28, vectors with atypical directions will return a similar distance measure, whereas the distance measured for similar vectors will increase. As the intersection points are calculated on the unit sphere but the distance measured is Euclidean, there is an inherent non-linearity in this method.

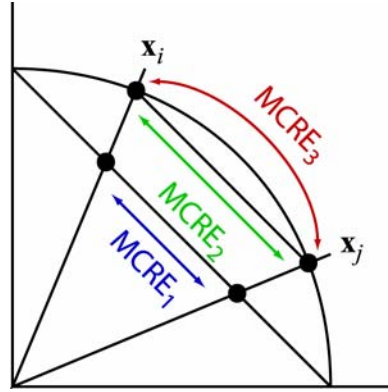


Figure 4.5. The three representations of the mean chromaticity error metric.

The author has determined that a third definition of the MCRE may be given: the distance between two points across the surface of the unit sphere. This is calculated simply as:

$$MCRE_3 = \cos^{-1} \left( \frac{\mathbf{x}_i \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|} \right) \quad (4.30)$$

where the arccosine returns an angle in radians. Figure 4.5 shows the three different MCRE metrics for two example 2D vectors.

To prevent confusion, in this work the MCRE employed is as given by Trahanias and Venetsanopoulos (Eq. 4.28).<sup>86</sup>

For images which are ultimately intended for human viewing, the CIELAB perceptual error metric is employed. Transforming the vector pair under consideration to LAB space (Section 2.8) the error is measured as the Euclidean distance between the two points:

$$LABE = \frac{\sum_{x=1}^N \sum_{y=1}^M \Delta E(x, y)}{NM} \quad (4.31)$$

where

$$\Delta E = \sqrt{(L^* - \bar{L}^*)^2 + (a^* - \bar{a}^*)^2 + (b^* - \bar{b}^*)^2} \quad (4.32)$$

Although there is some computational overhead in converting vectors from RGB to LAB, this method has the benefit of being a closer measure to the error perceived by the human eye than the other error metrics described. The LAB colourspace is preferred over LUV because in general machine vision techniques are concerned with comparison between off-camera images, i.e. reflected light sources. If the performance of a filter is based on screen appearance a more appropriate colourspace would be LUV, suitable for emitted light (Chapter 2).

A recent error measure introduced by Sangwine and Bardos<sup>91</sup> estimates noise in images using a pixel classification method based on the VMF. Given that a local window of pixels are rank-ordered using Eq. 4.8 and the  $L_2$  norm such that  $\mathbf{x}_1 \leq \mathbf{x}_2 \leq \dots \leq \mathbf{x}_N$ , if the pixel under consideration is  $\mathbf{x}_N$  it is labelled as noisy. The total noise measure is given as a percentage:

$$Err_{SB} = \frac{\sum_{x=1}^N \sum_{y=1}^M S_{xy}}{NM} \cdot 100 \quad (4.33)$$

where  $S_{xy}=1$  if classed as noisy,  $S_{xy}=0$  otherwise.

Whilst this measure may be applied to only a single image to ascertain how ‘noisy’ it is, without comparison between filtered and original images it cannot be considered a reasonable method of measuring a filter’s efficacy. It is only really a measure of an image’s smoothness; for instance an image with high spatial-frequency characteristics (such as one containing many edges) would return a high error score with this metric. Furthermore, it would not be able to detect the presence of Gaussian contamination, and as such can only be considered a useful measure for detecting noise which generates significant outliers.

Due to its popular usage in the literature, the NMSE is utilized as the primary error metric in this work, with the MCRE and LAB error included in places for comparative purposes.

## **4.7 PERFORMANCE OF DISCUSSED FILTERS**

To test the efficacy of the filters eight standard colour images from the Waterloo Repertoire image database were contaminated with noise in six different ways; 2% random impulse noise, 5% random impulse noise, Gaussian noise of variance 100, Gaussian noise of variance 1000 and two of mixed mode (2% impulse plus Gaussian variance 100 and 5% impulse plus Gaussian variance 1000). These levels and types of noise are used to permit comparison with established research which employs similar types and amounts of contamination. In the interests of brevity, and in view of the relatively low noise levels, results for only  $3 \times 3$  filter window sizes are given.

### **4.7.1 EXPERIMENT 1: 2% RANDOM IMPULSE NOISE**

Regarding the graphic results of this first experiment, some interesting observations may be made concerning the variation between the images, filters and noise metric used. First, let us consider the error measured for the unprocessed, contaminated images. For the NMSE results (Figure 4.6a) there is a marked difference between the images of the noise level measured; under 0.01 for ‘Clegg’ and ‘Serrano’ and over three times higher at nearly 0.03 for ‘Sail’, contrary to what one would expect for images contaminated with the same amount and type of noise. For the MCRE and LAB error results (Figures 4.6b and Figure 4.6c respectively) much more uniform variations are seen (less than 1 and 3 for the MCRE and LAB error respectively) compared to the errors observed for the filters. For the MCRE and LAB metrics, it would appear all of the filters tested worsen the error score, rather than improve it. Qualitative observations support this; the level of contamination is relatively low so the distortion introduced by application of a filter is visually worse than the presence of the original noise.



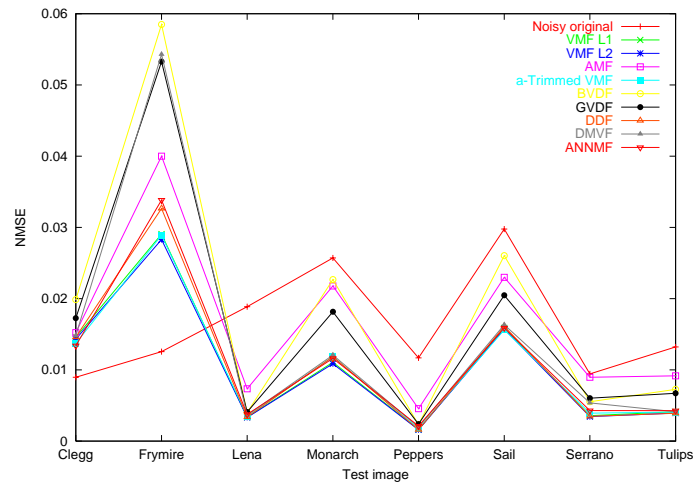


Figure 4.6a. NMSE for 2% impulse noise contamination.

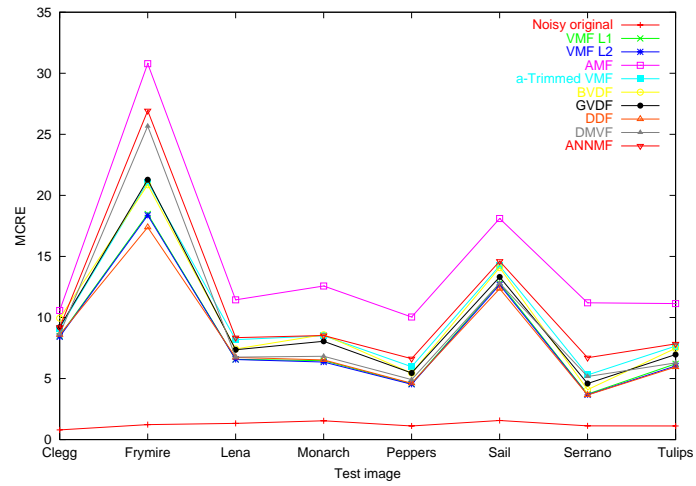


Figure 4.6b. MCRE for 2% impulse noise contamination.

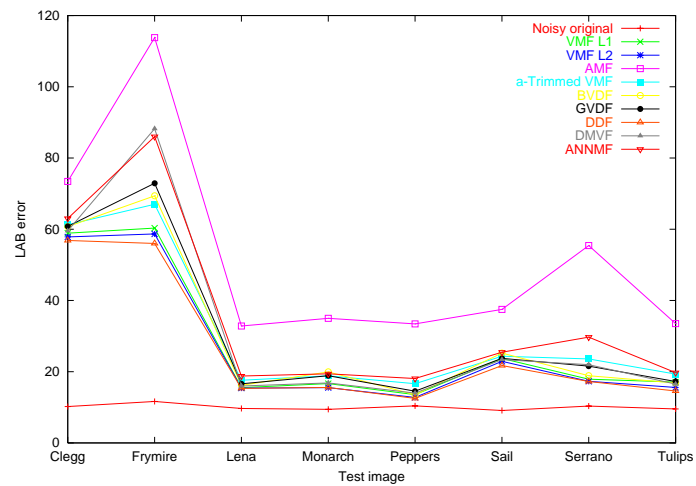


Figure 4.6c. LAB error for 2% impulse noise contamination.

	Average error scores for 2% impulse noise contamination to 5 significant figures.		
Filter	NMSE	MCRE	LAB
VMF ( $L_1$ )	0.010409	8.4138	27.954
VMF ( $L_2$ )	<b>0.010191</b>	8.3257	26.999
AMF	0.016255	14.484	51.859
$\alpha$ VMF	0.010422	10.004	31.073
BVDF	0.018273	9.7380	30.088
GDVF	0.016048	9.5293	30.784
DDF	0.010950	<b>8.2239</b>	<b>26.226</b>
DMVF	0.014082	9.6329	32.057
ANNMF	0.011147	11.095	35.028

**Table 4.1. Average error scores for 2% impulse noise contamination. The best scores are shown in bold.**

It is clear from all three figures that the performance of the filters varies more greatly between the test images than between the filters themselves; the worst performing being those, as expected, that have a large amount of high-frequency detail, especially the ‘Frymire’ image containing many edges.

No one filter clearly outperforms any other, but in general the averaging-type and directional filters perform poorly against the standard  $L_1$  and  $L_2$  norm VMF. The  $\alpha$ VMF performs consistently, but the DDF has the best average MCRE and LAB error scores, only trailing the standard VMF and  $\alpha$ VMF for the NMSE (Table 4.1).

	Average error scores for 5% impulse noise contamination to 5 significant figures		
Filter	NMSE	MCRE	LAB
VMF ( $L_1$ )	0.011266	8.8226	29.541
VMF ( $L_2$ )	0.011042	8.7448	28.579
AMF	0.020190	16.487	64.902
$\alpha$ VMF	<b>0.010948</b>	10.270	32.052
BVDF	0.019699	10.128	31.814
GDVF	0.016913	9.7988	32.001
DDF	0.011595	<b>8.6133</b>	<b>27.709</b>
DMVF	0.015322	10.029	33.724
ANNMF	0.011986	11.595	37.619

**Table 4.2 Average error scores for 5% impulse noise contamination. The best scores are shown in bold.**

## 4.7.2 EXPERIMENT 2: 5% RANDOM IMPULSE NOISE

The results of this experiment (shown in Figure 4.7a-c) show an almost identical trend to Expt. 1. All the filters continue to worsen the error score for the MCRE, but for the majority of the natural colour images ('Lena', 'Monarch', 'Peppers' and 'Tulips') all filters but the AMF improve the score for the LAB error.

As in Expt. 1, the DDF gives the best average error score with respect to the MCRE and LAB metrics (Table 4.2), followed closely by the standard VMF. For the average NMSE error the  $\alpha$ VMF has the lowest score, but the difference between this and the  $L_2$  norm VMF is very small ( $9.4 \times 10^{-6}$ )

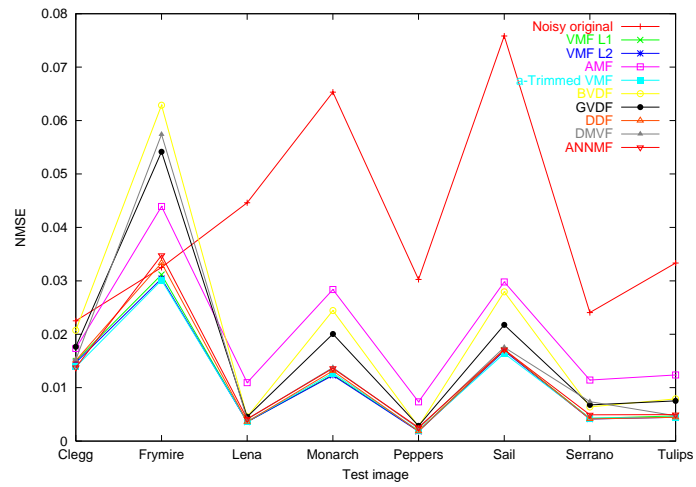


Figure 4.7a. NMSE for 5% impulse noise contamination.

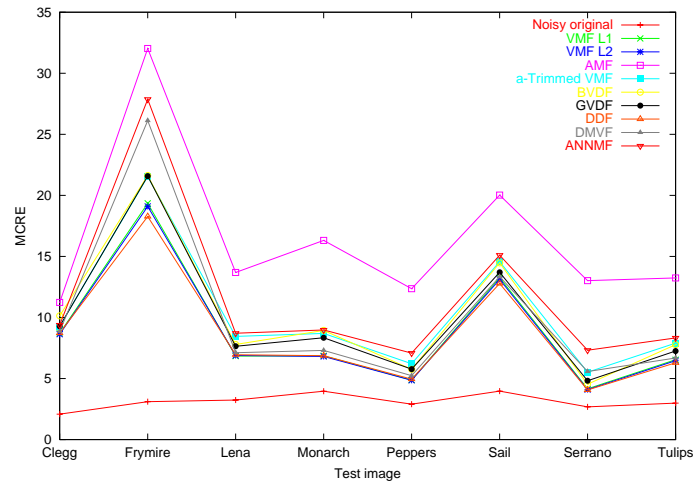


Figure 4.7b. MCRE for 5% impulse noise contamination.

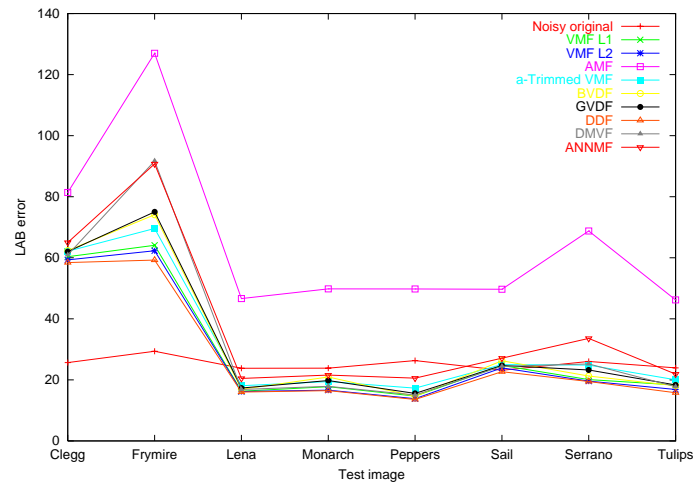


Figure 4.7c. LAB error for 5% impulse noise contamination.

### 4.7.3 EXPERIMENT 3: LIGHT GAUSSIAN NOISE ( $\sigma^2=100$ )

The results of each metric in this experiment are quite different to Expts. 1 and 2. The measured error for the original noise contaminated image is fairly uniform for the NMSE (Figure 4.8a) unlike the MCRE and LAB error (Figures 4.8b-c), contrary to the observation for Expts. 1 and 2. It can also be seen that the majority of the filters worsen the error score for the NMSE but not the MCRE or LAB error, dissimilar to the case for Expts. 1 and 2.

Also for this experiment the performance of the filters relative to each other is in agreement across the three error metrics; once more the BVDF and GVDF are the worst performing filters for the great majority of measurements, but the best performing are the averaging-type hybrid filters (ANNMF and  $\alpha$ VMF) as would be expected for Gaussian-type noise. The AMF also performs well for the natural images with the MCRE and LAB metrics, but somewhat poorly for the images of artificial origin with high-frequency content ('Clegg', 'Frymire' and 'Serrano').

The standard VMF is a solid middle-performer, but again the DDF offers an improvement due to its chromaticity preserving component; these filters seem especially suited to those images with high-frequency content when considering the NMSE and MCRE metrics.

The average error scores given in Table 4.3 support these observations; the BVDF and GVDF are clearly the worst performers, and the  $\alpha$ VMF and ANNMF are evidently the best filters for this type of noise.

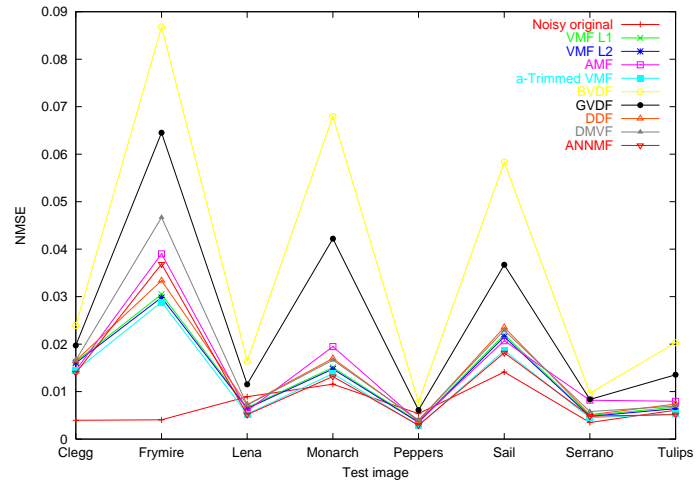


Figure 4.8a. NMSE for  $\sigma^2=100$  Gaussian noise contamination.

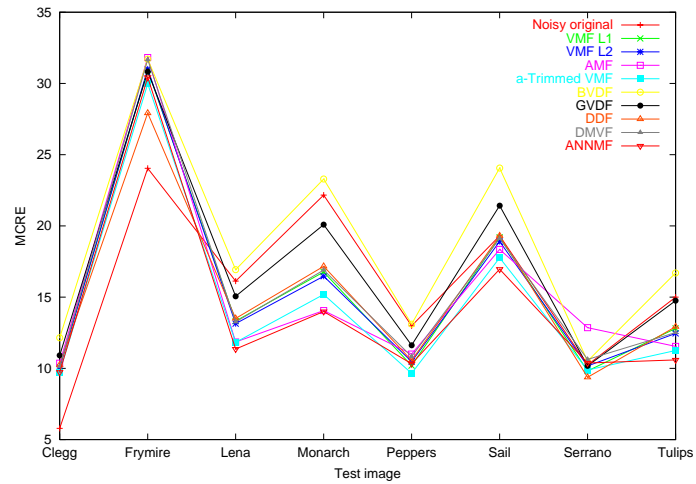


Figure 4.8b. MCRE for  $\sigma^2=100$  Gaussian noise contamination.

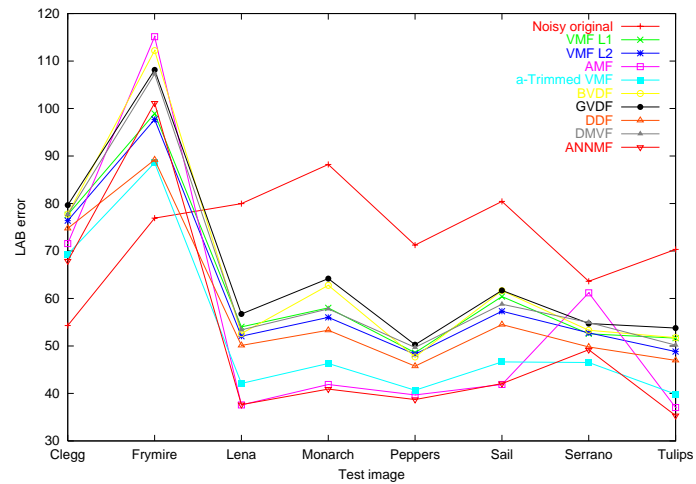


Figure 4.8c. LAB error for  $\sigma^2=100$  Gaussian noise contamination.

	Average error scores for $\sigma^2=100$ Gaussian noise contamination to 5 significant figures		
Filter	NMSE	MCRE	LAB
VMF ( $L_1$ )	0.013244	15.394	62.723
VMF ( $L_2$ )	0.012916	15.331	61.167
AMF	0.014944	15.231	55.744
$\alpha$ VMF	<b>0.011704</b>	14.409	52.499
BVDF	0.036402	18.557	64.964
GDVF	0.025337	16.858	66.144
DDF	0.014254	15.122	58.046
DMVF	0.015854	15.631	63.720
ANNMF	0.012531	<b>14.219</b>	<b>51.617</b>

**Table 4.3. Average error scores for  $\sigma^2=100$  Gaussian noise contamination. The best scores are shown in bold.**

#### 4.7.4 EXPERIMENT 4: MODERATE GAUSSIAN NOISE ( $\sigma^2=1000$ )

The results of Expt. 4 (Figure 4.9a-c) show a similar trend for the measured error of the original noise-contaminated image across the three metrics, particularly high for the ‘Monarch’ and ‘Sail’ images. For the MCRE and LAB results, all the filters show an improvement over the basic error score with the exception of the BVDF, once more the worst performing filter; the GDVF also returns poor results, but the standard VMF is now amongst the poorest as well, using the MCRE and LAB metrics.

As far as the best performing filters are concerned, the measurements of the three metrics are in broad agreement; the AMF is the leader under heavy Gaussian contamination, as discussed in Section 4.2.1, followed by the ANNMF and  $\alpha$ VMF. This is corroborated by the average errors given in Table 4.4.

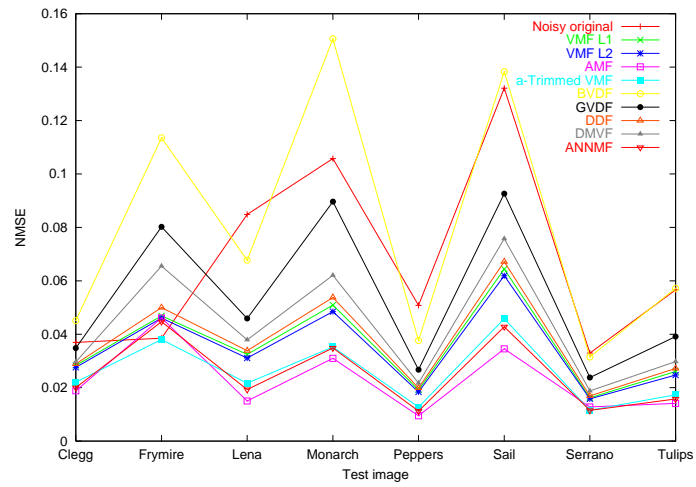


Figure 4.9a. NMSE for  $\sigma^2=1000$  Gaussian noise contamination.

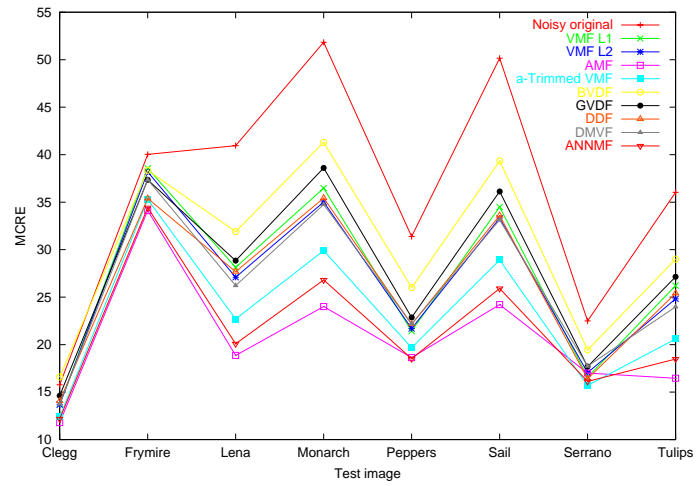


Figure 4.9b. MCRE for  $\sigma^2=1000$  Gaussian noise contamination.

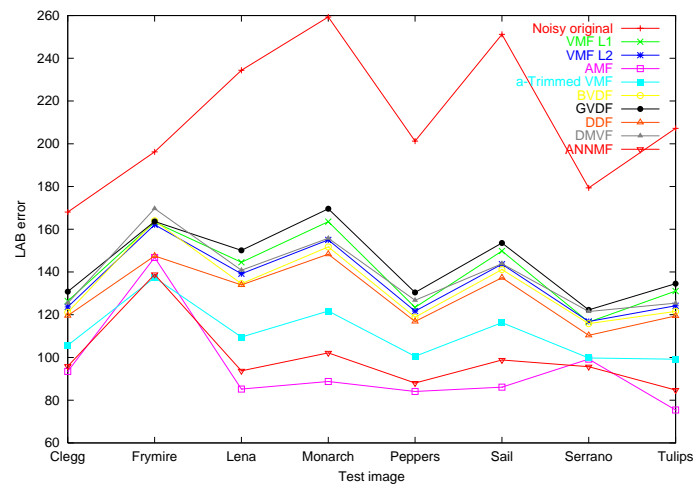


Figure 4.9c. LAB error for  $\sigma^2=1000$  Gaussian noise contamination.



	Average error scores for $\sigma^2=1000$ Gaussian noise contamination to 5 significant figures		
Filter	NMSE	MCRE	LAB
VMF ( $L_1$ )	0.035556	26.950	139.83
VMF ( $L_2$ )	0.034262	26.355	135.78
AMF	<b>0.022740</b>	<b>20.635</b>	<b>94.878</b>
$\alpha$ VMF	0.025557	23.143	111.24
BVDF	0.080212	30.239	133.58
GDVF	0.054102	27.902	144.35
DDF	0.037206	26.264	129.15
DMVF	0.042650	26.128	138.65
ANNMF	0.025019	21.566	99.739

Table 4.4. Average error scores for  $\sigma^2=1000$  Gaussian noise contamination. The best scores are shown in bold.

#### 4.7.5 EXPERIMENT 5: LIGHT MIXED NOISE (2% + $\sigma^2=100$ )

As the 2% impulse noise affects only a small proportion of pixels as opposed the Gaussian contamination which changes every pixel to some degree, it is to be expected that the error curves for Expt. 5 (Figure 4.10a-c) show more similarity to Expt. 3 than to Expt. 1. With the exception of the AMF whose performance deteriorates with the addition of the 2% impulse noise, all the filters perform almost identically to Expt. 3. The averaging-type filters perform the best, only bettered by the DDF for some images. Looking at the average scores in Table 4.5 the best filters across all metrics are the ANNMf and  $\alpha$ VMF.

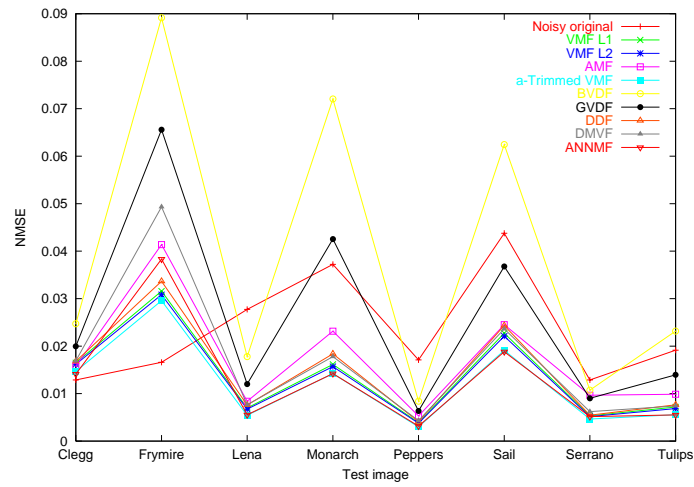


Figure 4.10a. NMSE for 2% impulse and  $\sigma^2=100$  Gaussian noise contamination.

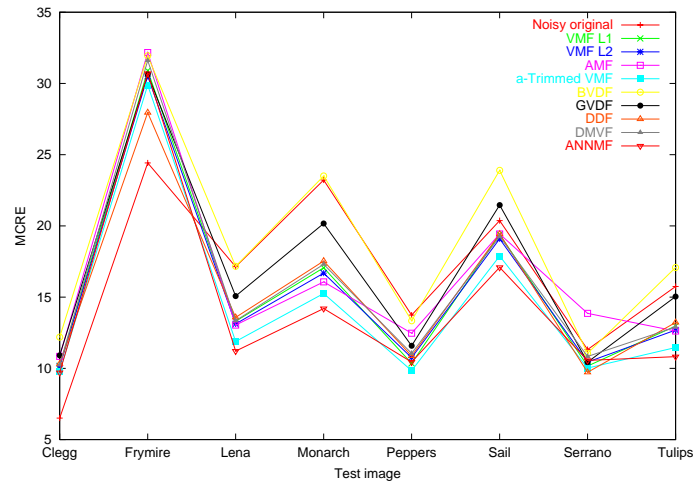


Figure 4.10b. MCRE for 2% impulse and  $\sigma^2=100$  Gaussian noise contamination.

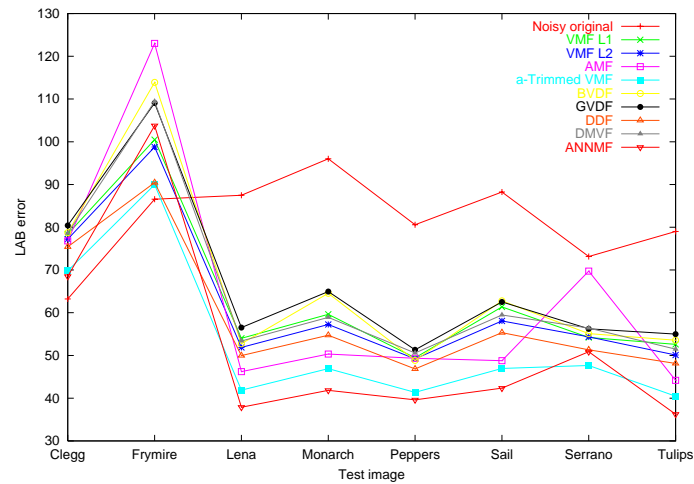


Figure 4.10c. LAB error 2% impulse and for  $\sigma^2=100$  Gaussian noise contamination.

	Average error scores for 2% impulse plus $\sigma^2=100$ Gaussian noise contamination to 5 significant figures		
Filter	NMSE	MCRE	LAB
VMF ( $L_1$ )	0.013791	15.543	63.764
VMF ( $L_2$ )	0.013412	15.427	62.105
AMF	0.017253	16.313	63.552
$\alpha$ VMF	<b>0.012025</b>	14.499	53.127
BVDF	0.038557	18.772	66.275
GDVF	0.025772	16.915	67.005
DDF	0.014747	15.328	59.023
DMVF	0.016651	15.815	64.784
ANNMF	0.013106	<b>14.341</b>	<b>52.636</b>

**Table 4.5. Average error scores for 2% impulse plus  $\sigma^2=100$  Gaussian noise contamination. The best scores are shown in bold.**

#### 4.7.6 EXPERIMENT 6: MODERATE MIXED NOISE (5% + $\sigma^2=1000$ )

As one would expect with heavier Gaussian contamination, the AMF performs better in this experiment than in Expt.5 as can be seen from Figures 4.11a-c. Overall, the best performing filter is the ANNMF, seconded by the AMF and  $\alpha$ VMF in all metrics but the NMSE where the  $\alpha$ VMF scores most highly, illustrated by the average error in Table 4.6. The DDF is a consistent middle performer along with the  $L_2$  VMF. The worst filters are once again the direction-based algorithms, with the notable exception of the BVDF for the LAB metric where it returns a similar score to the  $L_2$  VMF. Qualitative judgment is difficult to make; detail and features eroded by one filter are present in the other and vice versa; situations such as this highlight one reason to use different metrics to ascertain a filter's performance.

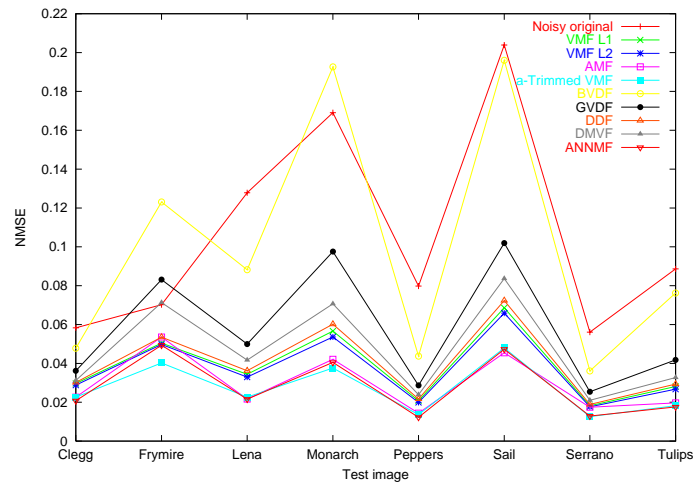


Figure 4.11a. NMSE for 5% impulse and  $\sigma^2=1000$  Gaussian noise contamination.

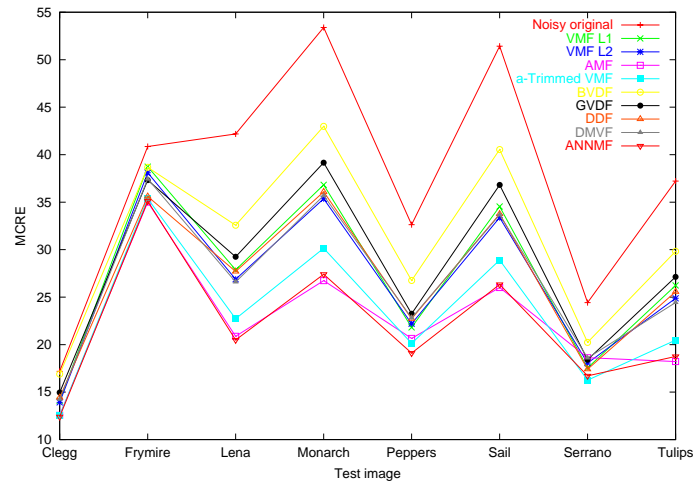


Figure 4.11b. MCRE for 5% impulse and  $\sigma^2=1000$  Gaussian noise contamination.

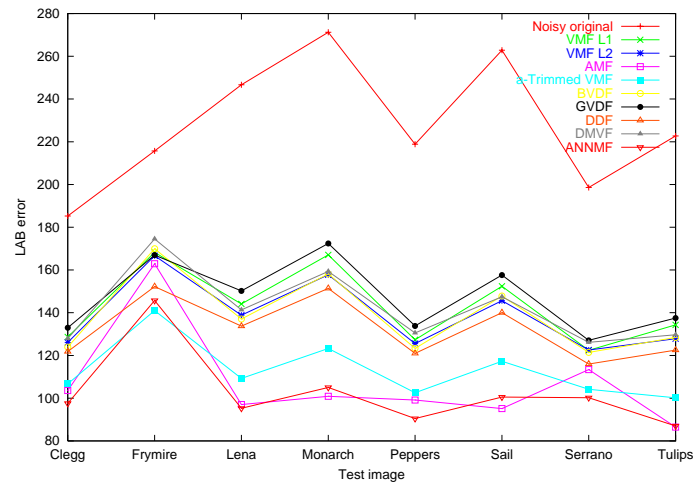


Figure 4.11c. LAB error 5% impulse and for  $\sigma^2=1000$  Gaussian noise contamination.

	Average error scores for 5% impulse plus $\sigma^2=1000$ Gaussian noise contamination to 5 significant figures		
Filter	NMSE	MCRE	LAB
VMF ( $L_1$ )	0.038349	27.206	143.05
VMF ( $L_2$ )	0.036926	26.597	138.86
AMF	0.029612	22.334	107.31
$\alpha$ VMF	<b>0.026936</b>	23.317	113.04
BVDF	0.100482	31.059	138.75
GDVF	0.058075	28.285	147.32
DDF	0.040299	26.659	132.32
DMVF	0.047019	26.687	142.11
ANNMF	0.027775	<b>22.039</b>	<b>102.72</b>

**Table 4.6. Average error scores for 5% impulse plus  $\sigma^2=1000$  Gaussian noise contamination. The best scores are shown in bold.**

## 4.7.7 DISCUSSION

Two observations are immediately clear when considering the results of these experiments; firstly that the results of a given metric do not necessarily correspond to a qualitative judgment and that when dealing with multivariate data it is important not to rely on one metric alone. Secondly, the class of image has a significant bearing on the performance of the filter; natural images require a different approach to those that are of artificial origin or otherwise have different statistics; images captured for industrial applications for instance.

The poor performance of the directional-type filters (BVDF, GVDF, DMVF) illustrates that intensity (vector length) is a more important feature than chromaticity (vector angle) in local areas. The BVDF was shown to be especially bad for Gaussian-contaminated images, due to pixels of similar low saturation colour but

different intensities all displaying entirely atypical directions after contamination; BVDF local window pixel selection then becomes virtually random. Venetsanopoulos and Trahanias found that BVDF outperformed the VMF for the MCRE but not the NMSE as one would expect.<sup>86</sup> However, these results are quite contrary to those shown here where the BVDF performs poorly in the MCRE metric for all noise models studied.

Plataniotis et al give results which show the DMVF outperforms the other vector filters described so far (VMF, BVDF, GVDF and DDF) for both Gaussian and impulsive noise using the ‘Lena’ test image.<sup>82</sup> At best the DMVF performs closely to the standard VMF but does not exceed it for any of the tests with the ‘Lena’ image, even though the noise models used by Plataniotis et al are practically identical to those employed here. It is difficult to reconcile why there is such a marked difference between the published work and that presented here.

The DDF performed well by comparison, especially for impulse contamination, so vector angle preservation clearly needs to be taken into account; however, the angle component of Eq. 4.20 is small compared to the intensity component, thus vector length is still the most important feature in this equation. Vardoulia et al found that the DDF performed poorly against the VMF in all their tests,<sup>77</sup> again disagreeing with the results shown here, although their work agrees that the BVDF is indeed a poor filter for generalized work. This highlights again that the context in which filters are tested should be carefully examined.

In all cases the  $L_2$  VMF outperformed the  $L_1$  VMF showing that in practice the  $L_2$  norm is effective against both types of noise, impulsive and Gaussian, at the expense of higher computational complexity. The  $\alpha$ VMF also shows itself to be worthwhile filter to employ when the noise characteristics may not be known, working effectively in both types of noise and performing especially in the mixed noise environment.

Although the ANNMF is not as effective at suppressing impulse-only noise as published results indicate<sup>28</sup>, its performance in Gaussian and mixed-noise

environments is obvious, without the extensive blurring associated with the AMF. Weighted filtering is clearly an avenue worthy of further study.

## **4.8 ARTIFICIAL NEURAL NETWORKS**

Artificial neural networks (ANNs) are generally described as mathematical models which attempt to mimic, to a greater or lesser extent, the massively parallel architecture of the brain. The human brain consists of roughly one hundred billion ( $10^{11}$ ) relatively simple processing units each with connections to ten thousand neighbouring neurons, an architecture which thus has over  $10^{15}$  information pathways. Each neuron's output depends on the weighted sum of its inputs; coupled with suitable learning rules to adjust connection weights and an appropriate simple output processing function, networks capable of highly complex non-linear decisions are possible using a topology of highly interconnected basic processing units similar to that found in the mammalian brain.

### **4.8.1 APPLICATION TO NOISE REMOVAL**

ANNs have been applied with great success to a number of areas in machine vision; edge detection,<sup>96,97</sup> image segmentation / texture analysis,<sup>60,98,99,100,101</sup> feature recognition,<sup>103,104,105,106</sup> image compression<sup>107,108</sup> and colour palette quantization.<sup>68</sup> Recently ANNs have been applied to the problem of greyscale image restoration<sup>109</sup> and noise removal.<sup>110</sup> Noise removal was carried out by feeding local window pixel data into a trained ANN which subsequently specified the value of the centre output pixel. It was found that the ANN was able to reject noise, both impulsive and Gaussian, to a greater extent than traditional averaging or median-type filters. The network topology used was a feedforward multilayer perceptron with error backpropagation, described overleaf. This serves as an introduction to the background of work undertaken in Chapter 7, where it is investigated whether this type of ANN may also be successfully applied to the problem of noise removal in vector-valued images.

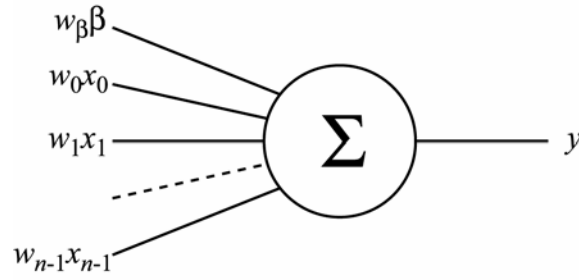


Figure 4.12. A simple perceptron.

### 4.8.1 THE PERCEPTRON

Pioneering work by McCulloch and Pitts in 1943<sup>92</sup> and later by Rosenblatt<sup>93,94</sup> lead to a mathematical definition of a neuron as a linear binary classifier, called a ‘perceptron’. The perceptron applied the weighted sum of its inputs to a threshold based classifier, usually a Heaviside step function:

$$y = \sum_{i=0}^{n-1} w_i x_i - \theta \quad (4.34)$$

where  $\mathbf{x}$  is the input vector,  $\mathbf{w}$  is the weight vector,  $y$  is the perceptron output and  $\theta$  is the threshold bias. Generally  $\theta$  is described as an additional perceptron input  $\beta$  fixed at 1 with a weight  $w_\beta$ :

$$y = \sum_{i=0}^{n-1} w_i x_i + w_\beta \beta \quad (4.35)$$

In this way the perceptron is able to classify a given input vector into one of two classes. This configuration is shown in Figure 4.12.

Training the perceptron is possible by moving the linear decision plane until there is no misclassification of input vectors. A simple weight update rule may be employed to alter the position of the linear discriminant plane by iterating through the training patterns:



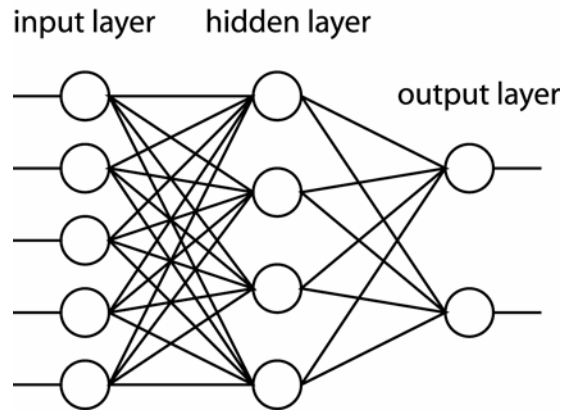


Figure 4.13. Example MLP architecture.

$$\begin{aligned} w_i(k+1) &= w_i(k) & y(k) &= Y(k) \\ w_i(k+1) &= w_i(k) + \eta[Y(k) - y(k)]x_i(k) & y(k) &\neq Y(k) \end{aligned} \quad (4.36)$$

where  $k$  represents the current iteration,  $Y$  represents the desired classification and  $\eta$  is a scaling factor. In other words, if the input vector is misclassified, the weight vector is altered in the direction of the correct classification.

However, the perceptron is only able to form a linear decision boundary which is clearly of little use in solving problems which require non-linear classification. By combining arrays of perceptrons into layers (see Figure 4.13) much more complex discriminant boundaries are possible, but the problem of systematic weight updating across the layers became a complex issue, and the development of artificial neural networks foundered for nearly two decades.

### 4.8.2 THE MULTILAYER BACK-PROPAGATION PERCEPTRON

Rumelhart et al found that the key to training multi-layer perceptrons (MLPs) was to utilize a continuous function rather than the traditional Heaviside step function.<sup>95</sup> A continuous function is necessary because it is the derivative which is used to update the weights of the neurons in the hidden layers; it is the rate of change of the neurons at their outputs which determines how their weights should be adapted. By far the most common function used is the sigmoid function, given by:

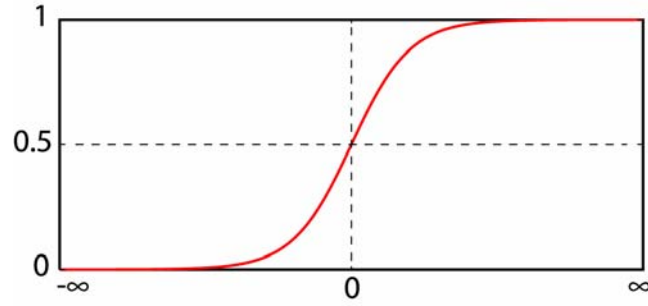


Figure 4.14. Sigmoid transfer function.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (4.37)$$

or:

$$f(x) = \tanh x \quad (4.38)$$

The first squashing function is used when neuron output should be clamped between 0 and 1, the second when the output should be between -1 and 1.

As well as conveniently limiting neuron output, these equations retain similarity to the original hard-limiting threshold function (see Figure 4.14) and have relatively simple derivative functions.

Given that the output of a neuron can be described as a function of its input  $o = f(i)$ , the error of a neuron at the output layer is calculated as:

$$\delta_k = f'(i_k)(t_k - o_k) \quad (4.39)$$

where  $o_k$  is the actual output of the neuron and  $t_k$  is the desired output target.

The derivative of the sigmoid function in Eq. 4.37 is simply:

$$f'(x) = \frac{1}{1 + e^{-x}} \left( 1 - \frac{1}{1 + e^{-x}} \right) \quad (4.40)$$

so Eq. 4.39 becomes:

$$\delta_k = (t_k - o_k)o_k(1 - o_k) \quad (4.41)$$

This error is propagated back to the hidden layer by adjusting the connection weights between the neurons in the hidden layer  $j$  and output layer  $k$ :

$$\overrightarrow{w_{kj}} = w_{kj} + \eta \delta_k o_j \quad (4.42)$$

where  $\overrightarrow{w_{kj}}$  is the updated weight and  $\eta$  is between 0 and 1, defined as the learning coefficient.

The error term for a neuron in the hidden layer is:

$$\delta_j = o_j(1 - o_j) \sum_{k=0}^{n_k} w_{kj} \delta_k \quad (4.43)$$

where the output neurons are labelled 0 to  $n_{k-1}$  and the  $n_k$ th neuron is the bias (or threshold) neuron.

The weight change for the connections between the input layer and hidden layer are calculated the same way as in Eq. 4.42:

$$\overrightarrow{w_{ji}} = w_{ji} + \eta \delta_j o_i \quad (4.44)$$

It is often the case that a weight  $w$  becomes stuck in a local minimum of the error surface; that is, any further small change in weight increases the error, although a somewhat larger increase in weight may decrease the error. For this reason a momentum term is often added to the weight update rule; for instance Eq. 4.42 becomes:

$$\overrightarrow{w_{kj}} = w_{kj} + \eta \delta_k o_j + \alpha \Delta w_{kj} \quad (4.45)$$

where  $\Delta w$  represent the previous weight change and  $\alpha$  is the momentum coefficient between 0 and 1. This addition makes it much more likely that local minima will be ignored and that the globally optimum solution will be reached.

It is important that neither  $\eta$  or  $\alpha$  is made too large, or the network will spend time oscillating around the minimum.

The overall fitness of the network is calculated as the sum-squared error of each training vector  $\mathbf{p}$ :

$$E_p = \frac{1}{2} \sum_{k=0}^{n_k-1} (t_{pk} - o_{pk})^2 \quad (4.46)$$

### 4.8.3 OTHER NETWORK TOPOLOGIES

The MLP falls into a class of ANN which requires supervised training. Many other network architectures are possible, such as Hopfield's autoassociative net<sup>111</sup> (often used as an example of content-addressable memory), Kohonen's self-organizing map<sup>112</sup> (popular in applications requiring unsupervised training), and Carpenter and Grossberg's Adaptive Resonance Theory<sup>113,114</sup> (viewed as powerful research models). The reader is directed to the references for further information on ANNs.

## 4.9 CONCLUDING REMARKS

This chapter has introduced the concept of noise removal using the local window, the traditional mean and median scalar filters and described the derivation of the vector median filter.

Popular extensions to the vector median have also been described, along with the metrics used to measure their efficacy, with a comprehensive comparison based on different noise models, metrics, and images, describing the advantages and disadvantages of each filter's design. Such a study has hitherto not been performed and provides details of each filter's strengths and weaknesses for a variety of image types not often apparent in the literature.

Finally, an introduction has been given to the subject of artificial neural networks and their application to image processing.

# Chapter 5: Extensions to the Vector Median Filter

## 5.1 EXPERIMENTS UNDERTAKEN

This chapter describes three original extensions to the vector median filter, the applicability of converting images to different colourspaces before processing, and a method for fast calculation of the VMF.

## 5.2 VECTOR MEDIAN FILTERING IN OTHER COLOURSPACES

As discussed in Chapter 2, a number of workers have used alternative colourspaces to facilitate or improve the results of their experiments in texture analysis and/or image segmentation. With the exception of the work by Vardavoulia et al<sup>77</sup>, no investigations into noise filtering using alternative colourspaces are known to have been performed. Vardavoulia's filter was designed with special regard for the HSV colourspace; the experiments done here investigate whether transformation of image data to other colourspaces aid the performance of the standard VMF.

The colourspaces used are Ohta's and LAB. The Ohta space was chosen as representative of colourspaces which decorrelate the image colour channels, and LAB space was used as an example of a colourspace attempting to mimic human colour perception.

As the VMF is recognized as a poor performer in Gaussian environments, experiments are restricted to images contaminated only with random impulsive noise. The level of contamination ranges from moderate to heavy (5% to 30% of total image pixels). The experiments were performed on the standard images from the Waterloo Repertoire Colorset image database. Each image was transformed to

the respective colour space for application of the VMF before the consequent inverse colour space transform; the noise measurements are conducted in RGB space.

### 5.2.1 RESULTS

The results using  $3 \times 3$  windows are given in Table 5.1a-d. The lowest error scores for each of the two metrics are shown in bold.

It can be seen that the transformation to either the Ohta or LAB colour space improves the error score in all instances barring the ‘Clegg’ image in the 5% and 30%. However, it can also be seen that where one alternative colour space gives a lower error score than RGB, the other colour space returns a worse score in a number of instances. It should perhaps be expected the lowest scores given by the LAB error metric are for the LAB colour space; the only exception being the ‘Serrano’ image in all experiments.

It should be noted that although the differences between the three colour spaces become more distinguished as the amount of noise increases, the improvement offered by performing the VMF in an alternative colour space cannot be considered particularly significant; in general the improvement is of the order of a few percent, with the greatest improvement only 14% (the ‘Sail’ image in Table 5.1d) using the NMSE metric in the LAB colour space. The processed ‘Sail’ images of Table 5.1d are shown in Figure 5.1, indicating the qualitative difference between processing in the RGB and LAB colour spaces.

Image	NMSE ( $\times 10^{-2}$ )				CIE LAB			
	No filter	RGB	Ohta	LAB	No filter	RGB	Ohta	LAB
Clegg	2.286	<b>1.483</b>	1.485	1.579	26.966	62.34	62.163	<b>61.790</b>
Frymire	3.497	3.046	<b>3.021</b>	3.029	31.421	62.868	62.487	<b>60.458</b>
Lena	4.479	0.350	<b>0.348</b>	0.359	24.966	16.714	16.608	<b>16.474</b>
Monarch	6.604	1.200	1.191	<b>1.169</b>	24.964	17.056	16.813	<b>15.749</b>
Peppers	2.977	0.230	0.228	<b>0.215</b>	27.554	15.239	15.077	<b>13.920</b>
Sail	7.693	1.684	1.676	<b>1.619</b>	24.314	24.980	24.732	<b>23.533</b>
Serrano	2.396	0.415	<b>0.411</b>	0.440	27.437	20.562	<b>20.403</b>	20.604
Tulips	3.376	0.444	0.434	<b>0.409</b>	25.182	17.691	17.374	<b>15.670</b>

**Table 5.1a. Error scores for 5% impulse noise contamination in alternative colourspaces.**

Image	NMSE ( $\times 10^{-2}$ )				CIE LAB			
	No filter	RGB	Ohta	LAB	No filter	RGB	Ohta	LAB
Clegg	4.666	1.581	<b>1.580</b>	1.676	54.755	65.615	65.508	<b>64.852</b>
Frymire	7.152	3.459	<b>3.425</b>	3.483	63.554	70.039	69.454	<b>67.877</b>
Lena	9.140	0.420	<b>0.415</b>	0.424	50.725	18.167	18.045	<b>17.836</b>
Monarch	13.355	1.408	1.396	<b>1.372</b>	50.319	18.905	18.614	<b>17.427</b>
Peppers	5.955	0.293	0.290	<b>0.280</b>	55.257	17.368	17.198	<b>16.044</b>
Sail	15.231	1.844	1.825	<b>1.763</b>	48.960	26.476	26.174	<b>24.926</b>
Serrano	4.730	0.520	<b>0.512</b>	0.559	54.918	24.087	<b>23.880</b>	24.489
Tulips	6.768	0.565	0.551	<b>0.511</b>	50.712	20.200	19.861	<b>18.131</b>

**Table 5.1b. Error scores for 10% impulse noise contamination in alternative colourspaces.**

Image	NMSE ( $\times 10^{-2}$ )				CIE LAB			
	No filter	RGB	Ohta	LAB	No filter	RGB	Ohta	LAB
Clegg	9.228	1.894	<b>1.892</b>	2.001	108.380	73.745	73.661	<b>73.076</b>
Frymire	14.137	4.612	<b>4.546</b>	4.715	125.724	87.678	86.442	<b>84.875</b>
Lena	18.053	0.679	<b>0.660</b>	0.692	100.631	22.220	21.923	<b>21.542</b>
Monarch	26.786	2.157	2.116	<b>2.007</b>	101.153	23.869	23.421	<b>21.851</b>
Peppers	12.052	0.576	<b>0.566</b>	0.589	111.531	23.532	23.232	<b>22.167</b>
Sail	30.601	2.451	2.395	<b>2.234</b>	97.417	30.753	30.270	<b>28.448</b>
Serrano	9.482	0.855	<b>0.837</b>	0.903	109.996	33.105	<b>32.643</b>	33.433
Tulips	13.686	0.938	0.921	<b>0.839</b>	101.390	26.057	25.686	<b>23.660</b>

**Table 5.1c. Error scores for 20% impulse noise contamination in alternative colourspace.**

Image	NMSE ( $\times 10^{-2}$ )				CIE LAB			
	No filter	RGB	Ohta	LAB	No filter	RGB	Ohta	LAB
Clegg	13.853	<b>2.448</b>	2.452	2.626	162.868	84.676	84.721	<b>84.573</b>
Frymire	21.568	6.393	<b>6.307</b>	6.972	190.735	111.693	110.144	<b>110.122</b>
Lena	27.335	1.278	<b>1.195</b>	1.268	150.891	28.366	27.520	<b>26.503</b>
Monarch	40.138	3.489	3.344	<b>3.145</b>	151.239	31.736	30.693	<b>27.714</b>
Peppers	18.062	1.055	<b>1.041</b>	1.238	167.523	32.079	31.770	<b>31.387</b>
Sail	46.179	3.734	3.551	<b>3.259</b>	146.667	37.416	36.373	<b>33.458</b>
Serrano	14.347	1.502	<b>1.472</b>	1.611	165.214	46.256	<b>45.496</b>	46.033
Tulips	20.614	1.642	1.596	<b>1.509</b>	152.622	34.916	34.356	<b>31.602</b>

**Table 5.1d. Error scores for 30% impulse noise contamination in alternative colourspace.**





Figure 5.1a. Original 'Sail' image.



Figure 5.1b. 'Sail' with 30% impulse noise.



Figure 5.1c. VMF in RGB colourspace.



Figure 5.1d. VMF in LAB colourspace.

---

### 5.2.2 CONCLUDING REMARKS

The merits bestowed by the alternative colourspace processing are significantly outweighed by the additional computational overhead in the examples studied here using the standard VMF and thus is not an effective preprocessing step in improving the efficacy of the VMF. However, this does not necessarily mean that utilizing other colourspace should be discounted for future work as an aid in the field of colour image noise filtering.

### 5.3 HYBRID MEAN MEDIAN FILTER (HMMF)

The experiments in Chapter 4 highlight the effectiveness of the  $\alpha$ -trimmed VMF for all noise types and can therefore be considered a good choice of general filter when the noise contamination is not known. As the  $\alpha$ VMF undergoes a rank order selection process before averaging it does not perform as well as the AMF due to the reduced data set passed to the averaging block. Conversely this method preserves more detail than the AMF for the same reason; less data smoothing occurs because the outliers have been rejected. The hybrid mean-median filter described here was designed for the same reason; good rejection of Gaussian noise whilst retaining image detail, and also acceptable performance in rejecting impulsive noise. The filter is defined as:

$$\mathbf{x}_{out} = \kappa \mathbf{x}_1 + (1 - \kappa) \mathbf{x}_2 \quad 0 \leq \kappa \leq 1 \quad (5.1)$$

where  $\mathbf{x}_1$  is the output vector of the AMF and  $\mathbf{x}_2$  is the output of the VMF. The parameter  $\kappa$  is chosen according to application and may be fixed or determined adaptively, by analyzing local image statistics for instance. By considering all vectors the HMMF will be able to smooth Gaussian noise more effectively than the  $\alpha$ VMF, but at the expense of preserving detail will not perform as well as the AMF if a mid-value of  $\kappa$  is chosen. In purely random impulse noise environments the HMMF will perform poorly against the VMF and  $\alpha$ VMF but will be superior to the AMF if again  $\kappa = 0.5$ . Clearly, if both types of noise are present and the rough quantities of each are known,  $\kappa$  can be tuned fairly intuitively;  $\kappa < 0.5$  produces a more favourable response when impulse noise is dominant,  $\kappa > 0.5$  when the noise is predominantly Gaussian.

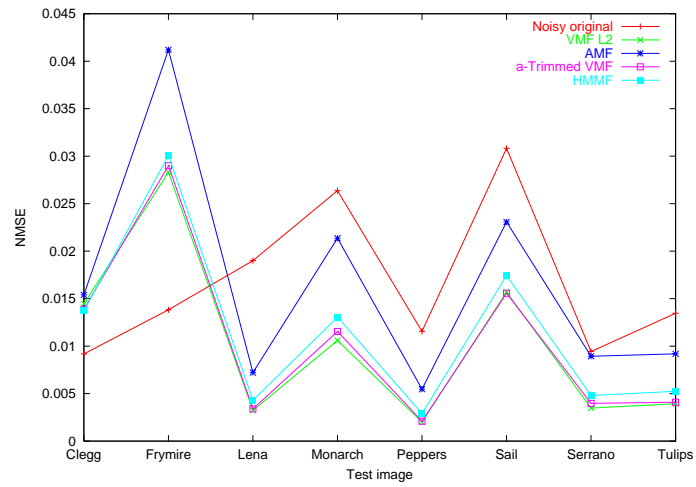


Figure 5.2a. NMSE for 2% impulse noise contamination.

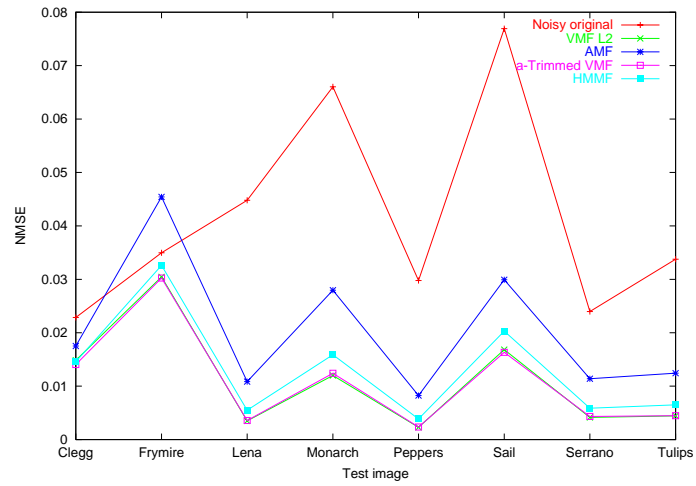


Figure 5.2b. NMSE for 5% impulse noise contamination.

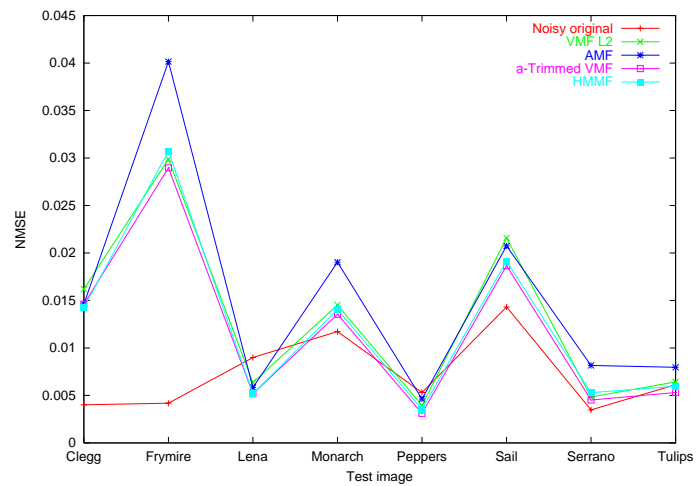


Figure 5.2c. NMSE for  $\sigma^2=100$  Gaussian noise contamination.

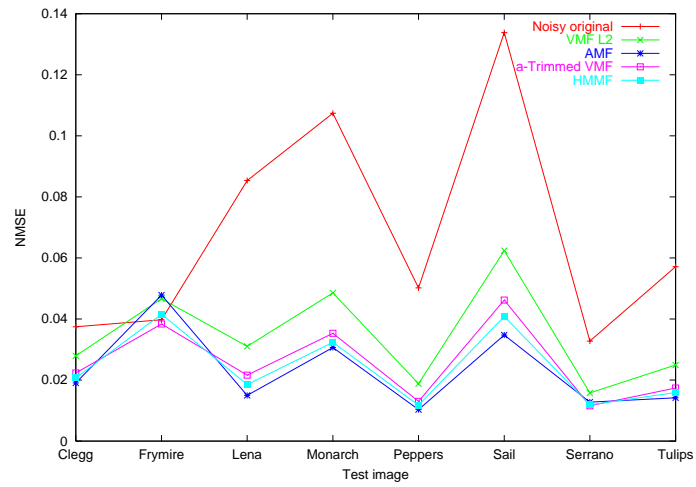


Figure 5.2d. NMSE for  $\sigma^2=1000$  Gaussian noise contamination.

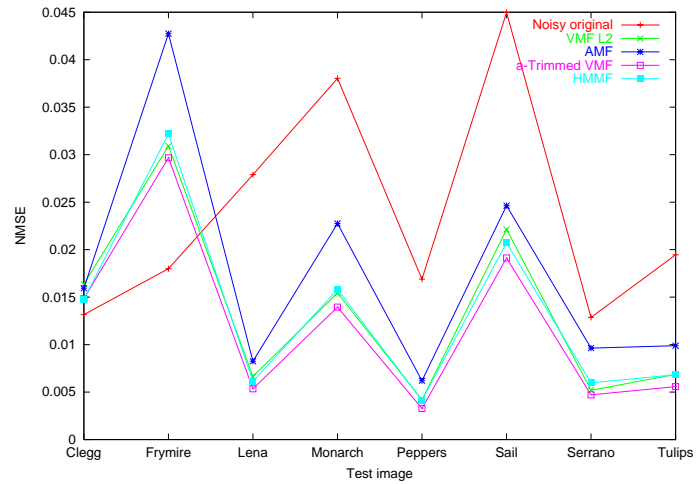


Figure 5.2e. NMSE for 2% impulse and  $\sigma^2=100$  noise contamination.

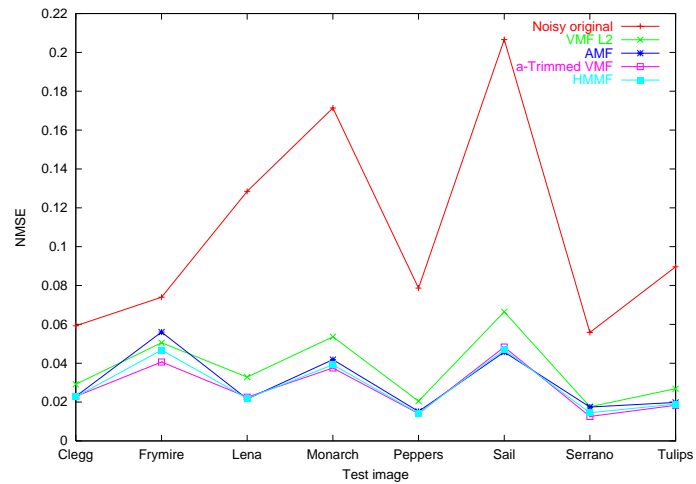


Figure 5.2f. NMSE for 5% impulse and  $\sigma^2=1000$  Gaussian noise contamination.

### 5.3.1 RESULTS

To illustrate this, experiments comparing the  $L_2$  VMF, AMF,  $\alpha$ VMF and the new HMMF were performed using the same noise models as Chapter 4. The parameter  $\kappa$  is fixed at 0.5. Again, for brevity only the results for the  $3 \times 3$  operator using the NMSE metric are shown (Figures 5.2a-f).

### 5.3.2 CONCLUDING REMARKS

As expected, the performance of the HMMF far surpasses that of the AMF in the mild impulsive noise tests (Figures 5.2a-b) but lags slightly behind the VMF and  $\alpha$ VMF. For light Gaussian noise ( $\sigma^2 = 100$ ) its performance is on par or only slightly lagging behind the  $\alpha$ VMF, illustrating its detail-preserving characteristics whilst suppressing Gaussian noise. In the light mixed noise experiment (Figure 5.3e) it performs similarly to the standard VMF, but in the heavy Gaussian noise and mixed noise experiments (Figures 5.2d and 5.2f) its effectiveness is clear, being the best performing filter in a number of instances.

The HMMF can be considered an alternative to the  $\alpha$ VMF and can be seen to provide an improvement in Gaussian contaminated environments; the ability to tune its capability adds to its versatility. It also shows some gains in computational speed over the  $\alpha$ VMF – there is no need to perform expensive sorting operations on the sum-of-distances of each vector. However, as with experiments carried out in Chapter 4, the strength of the  $\alpha$ VMF in all noise environments is evident.

## 5.4 SELECTIVE RANK ORDER FILTER (SROF)

When one considers the nature of impulse noise contamination, a pixel is either unchanged or assumes a random value that is probably an outlier when the local statistics are analyzed. The median filter suppresses impulse noise effectively, but does not discriminate between noisy and noise-free pixels; this results in a ‘softening’ of the image in noise-free areas.<sup>17</sup>

A better solution would be to leave a pixel unchanged unless local statistics show it is an outlier and therefore likely to be noisy. The filter described here is a modification of the VMF which examines the rank order of the pixel under consideration to decide if replacement is merited. The pixels in the local window are ordered in the same way as the standard VMF using an appropriate  $L_p$  norm distance metric such that pixels are labelled as described in Section 4.4.1. The output of the filter is then as follows:

$$\mathbf{x}_C = \begin{cases} \mathbf{x}_C & \mathbf{x}_C = \mathbf{x}_n \text{ for any } \mathbf{x}_n \quad n < \frac{N+1}{2}, n \in \mathbb{N} \\ \mathbf{x}_1 & \text{otherwise} \end{cases} \quad (5.2)$$

Where  $N$  is the number of pixels in the local window (assumed to be odd),  $\mathbf{x}_C$  is the pixel under consideration and  $\mathbf{x}_1$  is the pixel with the minimum output from the  $L_p$  norm metric, i.e. the vector median. Thus the pixel is left unchanged if it exists in the lower half of the rank order or replaced with the vector median if not; in this way outliers should be effectively rejected but noise-free pixels should be left as they are.

### 5.4.1 RESULTS

To demonstrate this filter’s effectiveness, results of experiments with very light to moderately heavy impulse noise (2% to 20% of total image pixels) are given. The inclusion of the  $\alpha$ VMF and DDF is to permit comparison with good impulse noise rejecting filters other than the VMF. Having been designed in mind to specifically

deal with impulse noise whilst retaining detail, this filter's performance with Gaussian noise is at best similar to that of the VMF; for this reason only results with impulse noise are shown, in Table 5.2a-f with the best figures emboldened.

### **5.4.2 CONCLUDING REMARKS**

It is clear from the results that the new SROF demonstrates a clear advantage over all the other well-known impulse noise rejecting filters using the  $3 \times 3$  operator. This is true all of all levels of contamination investigated which confirms its versatility for impulsive noise in general.

The efficacy of this filter is especially illustrated by the retention of much of the fine detail present in the artificial image 'Frymire', shown in Figure 5.3a-d.

This filter works by effectively 'ignoring' pixels if they fall on the lower end of the rank order; an alternative scheme could weight the vectors according to a heuristic or rule – such a scheme is described next in Section 5.5

	NMSE ( $\times 10^{-2}$ )				CIE LAB			
Image	VMF	DDF	$\alpha$ VMF	SROF	VMF	DDF	$\alpha$ VMF	SROF
Clegg	1.447	1.475	1.388	<b>1.121</b>	60.843	59.786	64.354	<b>37.064</b>
Frymire	2.83	3.275	2.900	<b>2.156</b>	59.073	56.634	67.980	<b>34.147</b>
Lena	0.323	0.328	0.34	<b>0.247</b>	15.863	15.790	18.166	<b>9.282</b>
Monarch	1.060	1.156	1.155	<b>0.757</b>	15.938	15.979	19.370	<b>8.710</b>
Peppers	0.200	0.190	0.209	<b>0.136</b>	14.058	13.361	18.095	<b>6.538</b>
Sail	1.569	1.608	1.556	<b>1.277</b>	23.912	22.707	25.420	<b>16.191</b>
Serrano	0.348	0.352	0.397	<b>0.245</b>	18.189	18.144	24.809	<b>10.371</b>
Tulips	0.394	0.391	0.409	<b>0.261</b>	16.367	15.371	20.294	<b>8.380</b>

**Table 5.2a. Error scores for 2% impulse noise contamination for the SROF using  $3 \times 3$  window.**

	NMSE ( $\times 10^{-2}$ )				CIE LAB			
Image	VMF	DDF	$\alpha$ VMF	SROF	VMF	DDF	$\alpha$ VMF	SROF
Clegg	1.483	1.524	1.405	<b>1.149</b>	62.340	61.404	65.251	<b>38.365</b>
Frymire	3.046	3.341	3.024	<b>2.285</b>	62.869	59.829	70.754	<b>36.541</b>
Lena	0.351	0.353	0.358	<b>0.260</b>	16.715	16.511	18.671	<b>9.742</b>
Monarch	1.200	1.283	1.242	<b>0.849</b>	17.056	16.898	19.979	<b>9.434</b>
Peppers	0.231	0.216	0.235	<b>0.155</b>	15.239	14.522	18.827	<b>7.297</b>
Sail	1.684	1.704	1.634	<b>1.346</b>	24.980	23.710	26.097	<b>16.816</b>
Serrano	0.416	0.417	0.434	<b>0.282</b>	20.563	20.445	26.129	<b>11.488</b>
Tulips	0.444	0.445	0.447	<b>0.285</b>	17.692	16.634	21.013	<b>9.042</b>

**Table 5.2b. Error scores for 5% impulse noise contamination for the SROF using  $3 \times 3$  window.**



Image	NMSE ( $\times 10^{-2}$ )				CIE LAB			
	VMF	DDF	$\alpha$ VMF	SROF	VMF	DDF	$\alpha$ VMF	SROF
Clegg	1.581	1.607	1.473	<b>1.217</b>	65.615	64.498	67.402	<b>40.862</b>
Frymire	3.459	3.645	3.272	<b>2.597</b>	70.040	66.316	76.065	<b>41.971</b>
Lena	0.421	0.418	0.401	<b>0.305</b>	18.167	17.867	19.697	<b>10.727</b>
Monarch	1.408	1.468	1.360	<b>0.999</b>	18.906	18.410	21.164	<b>10.782</b>
Peppers	0.293	0.277	0.279	<b>0.196</b>	17.368	16.597	20.067	<b>8.668</b>
Sail	1.845	1.831	1.742	<b>1.458</b>	26.476	25.028	27.113	<b>17.867</b>
Serrano	0.521	0.516	0.495	<b>0.352</b>	24.088	23.887	28.738	<b>13.701</b>
Tulips	0.565	0.548	0.517	<b>0.365</b>	20.201	18.974	22.387	<b>10.743</b>

**Table 5.2c. Error scores for 10% impulse noise contamination for the SROF using  $3 \times 3$  window.**

Image	NMSE ( $\times 10^{-2}$ )				CIE LAB			
	VMF	DDF	$\alpha$ VMF	SROF	VMF	DDF	$\alpha$ VMF	SROF
Clegg	1.894	1.909	1.649	<b>1.487</b>	73.746	72.335	72.593	<b>47.988</b>
Frymire	4.613	4.646	4.060	<b>3.626</b>	87.679	81.504	91.580	<b>56.620</b>
Lena	0.679	0.639	0.562	<b>0.525</b>	22.220	21.263	23.420	<b>14.090</b>
Monarch	2.157	2.089	1.830	<b>1.653</b>	23.870	22.605	25.529	<b>15.049</b>
Peppers	0.576	0.587	0.444	<b>0.427</b>	23.532	23.033	24.703	<b>13.487</b>
Sail	2.451	2.325	2.083	<b>1.982</b>	30.754	28.587	30.789	<b>21.305</b>
Serrano	0.856	0.835	0.711	<b>0.632</b>	33.105	32.610	36.531	<b>20.527</b>
Tulips	0.938	0.869	0.763	<b>0.655</b>	26.058	24.158	26.924	<b>15.220</b>

**Table 5.2d. Error scores for 20% impulse noise contamination for the SROF using  $3 \times 3$  window.**



Figure 5.3a. Original 'Frymire' image.

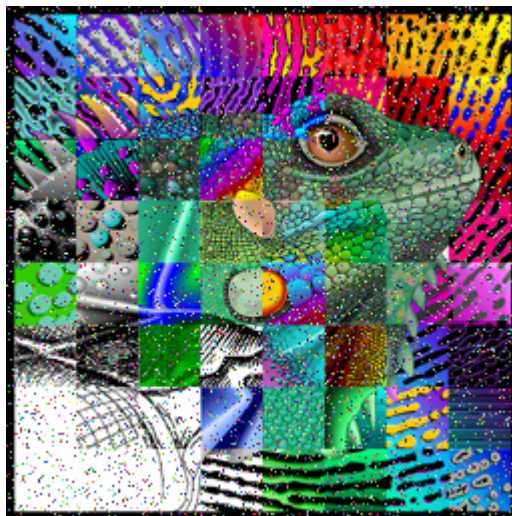


Figure 5.3b. 'Frymire' with 10% impulse noise.

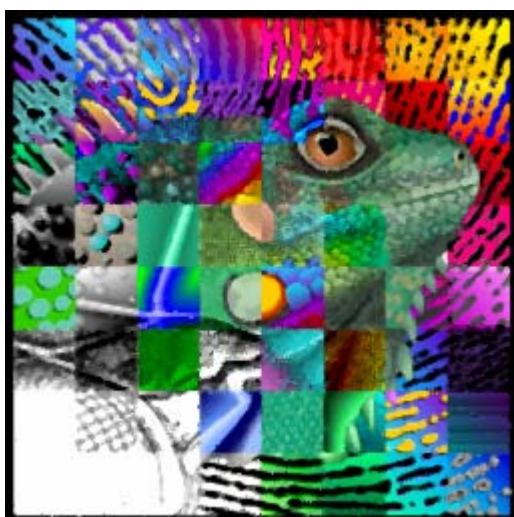


Figure 5.3c. Application of standard VMF.

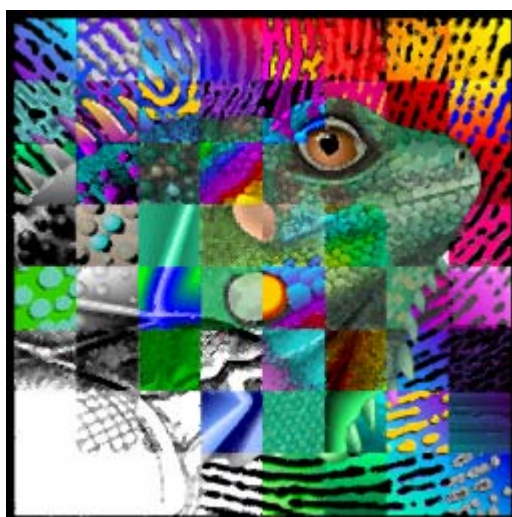


Figure 5.3d. Application of new SROF.

## 5.5 DISTANCE WEIGHTED VECTOR MEDIAN FILTER (DWVMF)

The concept of weighted median filters is well established<sup>115</sup> and a number of workers have given much attention to the development of weighted filters for colour image noise removal and enhancement<sup>28,78,115,116</sup> and also for applications such as image sequence coding.<sup>117</sup>

In one definition<sup>116</sup>, the weighting is achieved by duplication of samples inside the filter, such that with input sample set  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$  and corresponding weights  $\mathbf{w} = [w_1, w_2, \dots, w_n]$ , the output of the filter  $\mathbf{y}$  is:

$$\mathbf{y} = \text{median} [w_1 \diamond \mathbf{x}_1, w_2 \diamond \mathbf{x}_2, \dots, w_n \diamond \mathbf{x}_n] \quad (5.3)$$

where  $\diamond$  denotes weighting:

$$k \diamond \mathbf{x} = \overbrace{\mathbf{x}, \dots, \mathbf{x}}^{k \text{ times}} \quad (5.4)$$

Clearly, in this case the  $w_i$  are necessarily integral.

In contrast, the weighted distances are defined by Fotopoulos et al and Plataniotis et al respectively as:<sup>78,118</sup>

$$\mathbf{y} = \min_i w_i \sum_j |d_{ij}| \quad i \neq j \quad (5.5)$$

and

$$\mathbf{y} = \sum_j w_j \mathbf{x}_j \quad \sum_j w_j = 1 \quad (5.6)$$

where  $d_{ij}$  is the definition of distance given in Eq. 4.8 and there is no necessity for the weights  $w_i$  to be integral.

Here a variant filter is presented which modifies Eq. 5.5 by introducing a weighting function  $f(r_i)$  dependent on the Euclidean distance  $r_i$  between pixel  $i$  in the current

window and the pixel under consideration; it should be noted this occurs in the image plane, not in colour space.

In the next section the rationale of the new filter is described. Then in Section 5.5.2 results obtained are presented. In Section 5.5.3 the rationale is discussed in a wider context, and in Section 5.5.4 the value of the new technique is summarized.

### 5.5.1 THE RATIONALE OF THE NEW FILTER

It was stated in the previous section that a variant filter is introduced which incorporates a weighting function. In particular, Eq. 5.5 is modified by introducing a weighting function  $f(r_i)$  dependent on the Euclidean distance  $r_i$  between the spatial location of on-screen pixel  $i$  in the current window:

$$\mathbf{y} = \min_i f(r_i) \sum_j |d_{ij}| \quad i \neq j \quad (5.7)$$

$$r_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \quad (5.8)$$

where  $x_0$  and  $y_0$  represent the location of the centre pixel and  $f(r)$  is the weighting function chosen according to application. Possible choices include linear, piecewise linear and exponential, as shown in Figure 5.4.

The nonlinear example shown in Figure 5.4d exhibits unit weighting for distances up to  $a_1$ , and weighting proportional to distance from the centre up to a maximum value of  $a_2$ . This type of weighting might be expected to help with discretisation errors such as those which become apparent on measuring the shifts introduced by median filters<sup>119</sup>, though this prediction has not been checked explicitly here.

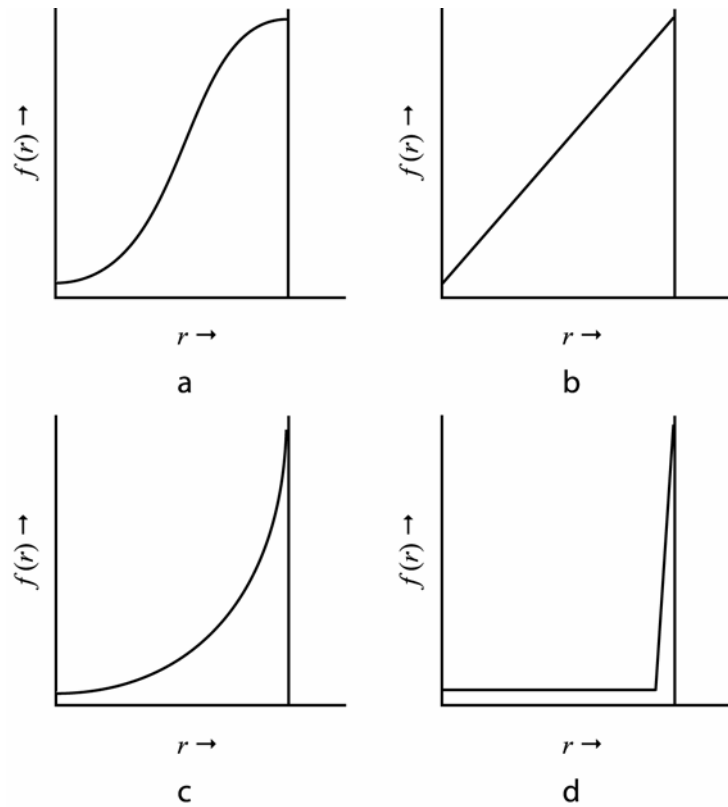


Figure 5.4. Examples of weighting functions.

One point about the weighting functions used in this work should be made clear. These are necessarily increasing functions which are largest at large distances. This is useful for continuity reasons, so that the limitations of the window support region do not have too great an effect on the properties of the operation: indeed, if the function value went to infinity at the boundary of the support region, increasing the size of the window would give no observable effect. (It is of course necessary to remember that the most relevant function values are those for small  $r_i$ , as Eq. 5.7 takes the *minimum* value as the output value.)

Clearly the type of weighting function used and the amount of weighting applied will vary with the type of application and desired results, but here it is shown that even in its basic form, this variant on the weighted median filter combines the detail-preserving properties of a small operator with the greater noise-averaging properties of a large operator.

It is intended that this should be achieved by making use of the increased generality of the new filter, which involves weights  $w_i$  that depend on the position of pixel  $i$  within the window (albeit pixels which have equal separation distances from the centre will have equal weights). In particular, pixels which are closer together should match each other's intensity and colour more closely, and in a general sense should be more relevant and able to preserve the detailed structure of the original picture. Indeed, the power spectra of natural images have been shown to follow a  $1/f^2$  slope,<sup>121</sup> and it has also been noted that correlation between pixel pairs falls as the distance between them increases.<sup>122</sup>

On the other hand, when there is considerable noise in the image, local averaging must be instituted over a greater region, and it then becomes useful to apply a weighting effect which brings in more pixels, so the functional value has to be decreased at larger distances, again in accordance with Eq. 5.7.

At this point we must recognise that there is a tradeoff between improved noise averaging capability and fidelity to the detail of the original image. This tradeoff can be adjusted by varying the weighting function. The question then becomes one of finding the optimum weighting function, and what criterion to use in order to achieve this. The weighting functions that seemed most basic and worthy of testing were the simple piecewise linear one of Figure 5.4b, and the square law function of Figure 5.4c, represented by Eq. 5.9 and Eq. 5.10, respectively:

$$f(r) = 1 + \frac{r}{\alpha} \quad (5.9)$$

$$f(r) = 1 + \frac{r^2}{\beta} \quad (5.10)$$

where  $r$  is defined in Eq. 5.8 and  $\alpha, \beta$  are arbitrarily chosen constants.

The criterion function that we adopted was the commonly used NMSE function. This would clearly give high values if too little weight were given to widely separated pixels (too little averaging to reduce noise) or too much weight were given to widely separated pixels (too little fidelity to the original image signal).



Figure 5.5a. Natural colour scene 1.



Figure 5.5b. Natural colour scene 2.

Clearly, if the right functional form is adopted, a central minimum should be observable in the variation of NMSE with pixel separation weight.

### 5.5.2 RESULTS

To demonstrate the efficacy of the new filter, the two natural test images (Figure 5.5) were contaminated with light to heavy noise (10%, 20%, 30% and 40% impulsive noise) and a comparison made between the standard L2 norm VMF and Distance Weighted VMF (DWVMF) for window sizes  $3 \times 3$  to  $9 \times 9$  pixels. ‘Circular’ windows were used for both operators, and the metric used for noise removal was the normalised mean square error measure (NMSE); this has the advantage of permitting comparison with earlier published material.<sup>28</sup>

In the trials simple linear and quadratic weighting functions  $f$  were employed, defined as in Eq. 5.9 and Eq. 5.10. These functions were chosen because of their computational simplicity, and they ensure values above unity for all the pixel weights. First the distance weighted VMF with linear weighting function  $f$  is considered. It can be seen that the noise rejection capability is bounded by two values; as  $\alpha \rightarrow 0$  the filter has no appreciable effect, and as  $\alpha \rightarrow \infty$  the filter approximates to the response of the normal VMF. In practice  $\alpha$  does not have to become very large before this occurs (Figure 5.6a–b). The trade-off is essentially between effective noise rejection and retention of detail.

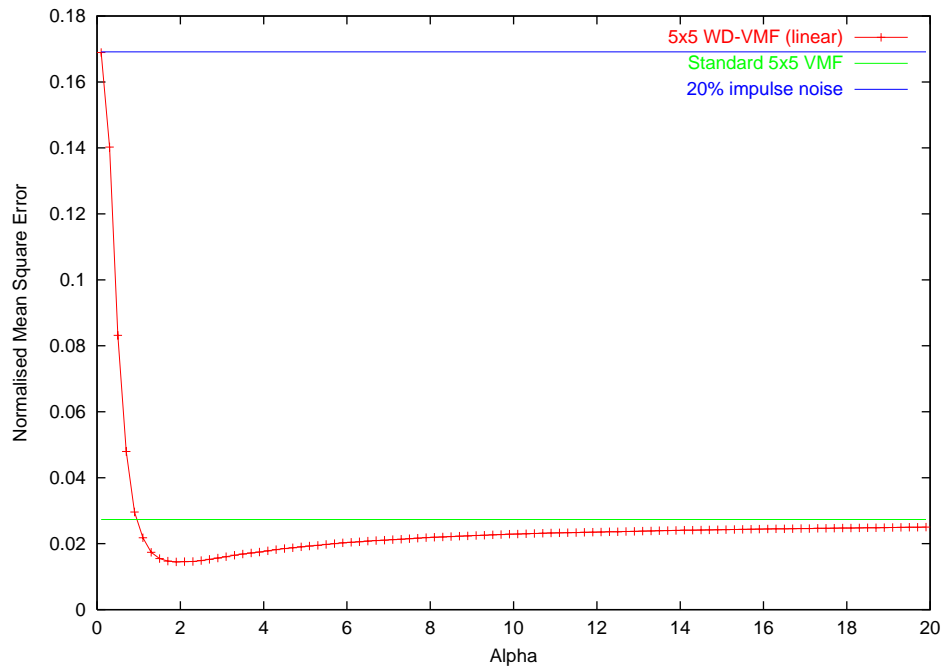


Figure 5.6a. Results of  $5 \times 5$  application of DWVMF filter for a range of parameter values using linear weighting function. NMSE of noisy image and  $5 \times 5$  standard vector median filtered image are shown for comparison.

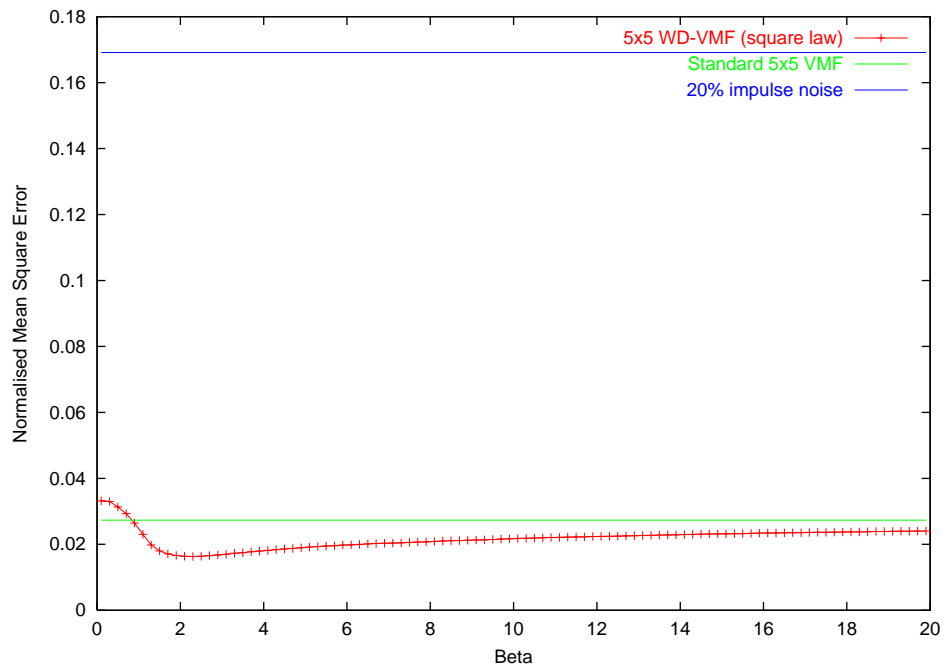


Figure 5.6b. Results of  $5 \times 5$  application of DWVMF filter for a range of parameter values using square law weighting function. NMSE of noisy image and  $5 \times 5$  standard vector median filtered image are shown for comparison.



Optimum values of  $\alpha$  were ascertained by taking the piecewise derivative of the NMSE results with respect to  $\alpha$ . The optimum results obtained for each noise model and window size are shown in the tables in Tables 5.3 – 5.10. The best results are shown in bold. For brevity only the results for the linear operator are shown in these tables, but both weighting functions are compared in Tables 5.11 – 5.12.

In each case, the linear DWVMF produces a significantly better NMSE than the standard VMF; in general, the size of operator that gives the optimum results increases with the amount of noise contamination, as indeed does the value of  $\alpha$  or  $\beta$ . This is to be expected, as at higher noise levels the influence of the larger neighbourhood should be emphasised to increase noise rejection.

Considering Figures 5.7 and 5.8, it is interesting to note that when the DWVMF is employed using smaller windows (Figures 5.7c–d, 5.8c–d) a reasonable level of high-frequency detail is still present at the expense of some smoothing and some clusters of noisy pixels which failed to be removed, whereas the larger operators (Figures 5.7e–f, 5.8e–f) preserved the high-frequency information much better and removed the noise more effectively in areas exhibiting similar features, such as the trees, bushes and grass areas, but performed very poorly at object boundaries and highly detailed areas, such as the facial features and clothing of the figures in the foreground in both test images. This is due to a lack of a representative dataset at those portions of the image which have highly varying statistics, i.e. at object boundaries which will have rapidly changing intensity and chromatic properties. Note that in all cases the retention of original image detail is better than for the standard VMF images (Figure 5.7b, 5.8b).

The case is somewhat different for the square law DWVMF. For images with low noise contamination, results are similar to the linear law weighting function; indeed for the least noisy images (10% impulse noise) they are better, with the exception of the  $3 \times 3$  operator. However, for experiments on noisier images, the linear law weighting function was definitely superior – in fact, the square law DWVMF returns a worse noise rejection measure than the standard VMF.

10% impulse noise, NMSE = $8.440 \times 10^{-2}$			
	NMSE		
Window size	Standard VMF	DWVMF	$\alpha$
$3 \times 3$	$1.624 \times 10^{-2}$	<b><math>8.019 \times 10^{-3}</math></b>	1.6
$5 \times 5$	$2.573 \times 10^{-2}$	$9.258 \times 10^{-3}$	1.2
$7 \times 7$	$3.175 \times 10^{-2}$	$1.036 \times 10^{-2}$	1.0
$9 \times 9$	$3.910 \times 10^{-2}$	$1.186 \times 10^{-2}$	1.0

**Table 5.3. Optimum value of  $\alpha$  for the DWVMF against the standard VMF for Figure 2(a) contaminated with 10% impulse noise. The best value is shown in bold.**

10% impulse noise, NMSE = $6.680 \times 10^{-2}$			
	NMSE		
Window size	Standard VMF	DWVMF	$\alpha$
$3 \times 3$	$1.103 \times 10^{-2}$	<b><math>5.047 \times 10^{-3}</math></b>	1.7
$5 \times 5$	$1.788 \times 10^{-2}$	$5.811 \times 10^{-3}$	1.3
$7 \times 7$	$2.297 \times 10^{-2}$	$6.865 \times 10^{-3}$	1.2
$9 \times 9$	$3.061 \times 10^{-2}$	$8.574 \times 10^{-3}$	1.2

**Table 5.4. Optimum value of  $\alpha$  for the DWVMF against the standard VMF for Figure 2(b) contaminated with 10% impulse noise. The best value is shown in bold.**

20% impulse noise, NMSE = $1.691 \times 10^{-1}$			
	NMSE		
Window size	Standard VMF	DWVMF	$\alpha$
$3 \times 3$	$1.943 \times 10^{-2}$	<b><math>1.321 \times 10^{-2}</math></b>	3.1
$5 \times 5$	$2.735 \times 10^{-2}$	$1.456 \times 10^{-2}$	2.1
$7 \times 7$	$3.291 \times 10^{-2}$	$1.544 \times 10^{-2}$	1.7
$9 \times 9$	$3.973 \times 10^{-2}$	$1.745 \times 10^{-2}$	1.6

**Table 5.5. Optimum value of  $\alpha$  for the DWVMF against the standard VMF for Figure 2(a) contaminated with 20% impulse noise. The best value is shown in bold.**

20% impulse noise, NMSE = $1.322 \times 10^{-1}$			
	NMSE		
Window size	Standard VMF	DWVMF	$\alpha$
$3 \times 3$	$1.403 \times 10^{-2}$	<b><math>9.233 \times 10^{-3}</math></b>	3.7
$5 \times 5$	$1.980 \times 10^{-2}$	$9.849 \times 10^{-3}$	2.3
$7 \times 7$	$2.471 \times 10^{-2}$	$1.083 \times 10^{-2}$	2.0
$9 \times 9$	$3.215 \times 10^{-2}$	$1.339 \times 10^{-2}$	1.9

**Table 5.6. Optimum value of  $\alpha$  for the DWVMF against the standard VMF for Figure 2(b) contaminated with 20% impulse noise. The best value is shown in bold.**

30% impulse noise, NMSE = $2.511 \times 10^{-1}$			
	NMSE		
Window size	Standard VMF	DWVMF	$\alpha$
$3 \times 3$	$2.637 \times 10^{-2}$	$2.186 \times 10^{-2}$	6.1
$5 \times 5$	$2.970 \times 10^{-2}$	<b><math>1.960 \times 10^{-2}</math></b>	3.1
$7 \times 7$	$3.509 \times 10^{-2}$	$2.075 \times 10^{-2}$	2.9
$9 \times 9$	$4.167 \times 10^{-2}$	$2.322 \times 10^{-2}$	2.6

**Table 5.7. Optimum value of  $\alpha$  for the DWVMF against the standard VMF for Figure 2(a) contaminated with 30% impulse noise. The best value is shown in bold.**

30% impulse noise, NMSE = $1.977 \times 10^{-1}$			
	NMSE		
Window size	Standard VMF	DWVMF	$\alpha$
$3 \times 3$	$1.920 \times 10^{-2}$	$1.544 \times 10^{-2}$	6.8
$5 \times 5$	$2.237 \times 10^{-2}$	<b><math>1.409 \times 10^{-2}</math></b>	3.6
$7 \times 7$	$2.693 \times 10^{-2}$	$1.503 \times 10^{-2}$	3.2
$9 \times 9$	$3.471 \times 10^{-2}$	$1.784 \times 10^{-2}$	2.7

**Table 5.8. Optimum value of  $\alpha$  for the DWVMF against the standard VMF for Figure 2(b) contaminated with 30% impulse noise. The best value is shown in bold.**

40% impulse noise, NMSE = $3.337 \times 10^{-1}$			
	NMSE		
Window size	Standard VMF	DWVMF	$\alpha$
$3 \times 3$	$4.319 \times 10^{-2}$	$3.979 \times 10^{-2}$	8.4
$5 \times 5$	$3.580 \times 10^{-2}$	$2.767 \times 10^{-2}$	5.4
$7 \times 7$	$3.930 \times 10^{-2}$	<b><math>2.704 \times 10^{-2}</math></b>	4.6
$9 \times 9$	$4.572 \times 10^{-2}$	$2.882 \times 10^{-2}$	3.9

**Table 5.9. Optimum value of  $\alpha$  for the DWVMF against the standard VMF for Figure 2(a) contaminated with 40% impulse noise. The best value is shown in bold.**

40% impulse noise, NMSE = $2.646 \times 10^{-1}$			
	NMSE		
Window size	Standard VMF	DWVMF	$\alpha$
$3 \times 3$	$3.094 \times 10^{-2}$	$2.760 \times 10^{-2}$	8.7
$5 \times 5$	$2.795 \times 10^{-2}$	<b><math>2.071 \times 10^{-2}</math></b>	5.4
$7 \times 7$	$3.201 \times 10^{-2}$	$2.103 \times 10^{-2}$	4.6
$9 \times 9$	$3.938 \times 10^{-2}$	$2.380 \times 10^{-2}$	4.3

**Table 5.10. Optimum value of  $\alpha$  for the DWVMF against the standard VMF for Figure 2(b) contaminated with 40% impulse noise. The best value is shown in bold.**

		10% noise		20% noise	
Window Size	Weighting Function	NMSE	$\alpha, \beta$	NMSE	$\alpha, \beta$
$3 \times 3$	Linear	<b><math>8.019 \times 10^{-3}</math></b>	1.6	<b><math>1.321 \times 10^{-2}</math></b>	3.1
	Square	$8.654 \times 10^{-3}$	1.7	$1.759 \times 10^{-2}$	3.8
$5 \times 5$	Linear	$9.258 \times 10^{-3}$	1.2	$1.456 \times 10^{-2}$	2.1
	Square	$9.136 \times 10^{-3}$	1.1	$1.621 \times 10^{-2}$	2.2
$7 \times 7$	Linear	$1.036 \times 10^{-2}$	1.0	$1.544 \times 10^{-2}$	1.7
	Square	$1.022 \times 10^{-2}$	0.9	$1.649 \times 10^{-2}$	2.1
$9 \times 9$	Linear	$1.186 \times 10^{-2}$	1.0	$1.745 \times 10^{-2}$	1.6
	Square	$1.116 \times 10^{-2}$	0.7	$1.736 \times 10^{-2}$	1.9

**Table 5.11a. Optimum performance of the linear and square law DWVMF for Figure 5.5a. for 10% and 20% impulse noise. The best value is shown in bold. For the linear case, the relevant parameter is  $\alpha$ , while for the square law case the relevant parameter is  $\beta$ .**

		30% noise		40% noise	
Window Size	Weighting Function	NMSE	$\alpha, \beta$	NMSE	$\alpha, \beta$
$3 \times 3$	Linear	$2.186 \times 10^{-2}$	6.1	$3.979 \times 10^{-2}$	8.4
	Square	$3.293 \times 10^{-2}$	8.9	$6.291 \times 10^{-2}$	14.3
$5 \times 5$	Linear	<b><math>1.960 \times 10^{-2}</math></b>	3.1	$2.767 \times 10^{-2}$	5.4
	Square	$2.622 \times 10^{-2}$	7.1	$4.014 \times 10^{-2}$	17.2
$7 \times 7$	Linear	$2.075 \times 10^{-2}$	2.9	<b><math>2.704 \times 10^{-2}</math></b>	4.6
	Square	$2.435 \times 10^{-2}$	6.7	$3.370 \times 10^{-2}$	16.0
$9 \times 9$	Linear	$2.322 \times 10^{-2}$	2.6	$2.882 \times 10^{-2}$	3.9
	Square	$2.671 \times 10^{-2}$	4.6	$3.431 \times 10^{-2}$	14.5

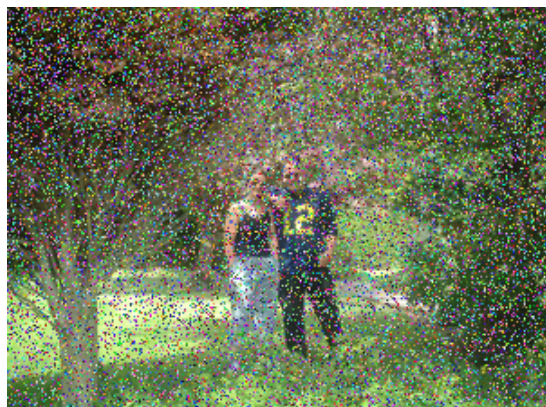
**Table 5.11b. Optimum performance of the linear and square law DWVMF for Figure 5.5a. for 30% and 40% impulse noise. The best value is shown in bold. For the linear case, the relevant parameter is  $\alpha$ , while for the square law case the relevant parameter is  $\beta$ .**

		10% noise		20% noise	
Window Size	Weighting Function	NMSE	$\alpha, \beta$	NMSE	$\alpha, \beta$
$3 \times 3$	Linear	<b><math>5.047 \times 10^{-3}</math></b>	1.7	<b><math>9.233 \times 10^{-3}</math></b>	3.7
	Square	$5.255 \times 10^{-3}$	2.0	$1.135 \times 10^{-2}$	4.1
$5 \times 5$	Linear	$5.811 \times 10^{-3}$	1.3	$9.849 \times 10^{-3}$	2.3
	Square	$5.746 \times 10^{-3}$	1.5	$1.059 \times 10^{-2}$	2.7
$7 \times 7$	Linear	$6.865 \times 10^{-3}$	1.2	$1.083 \times 10^{-2}$	2.0
	Square	$6.425 \times 10^{-3}$	1.4	$1.103 \times 10^{-2}$	2.5
$9 \times 9$	Linear	$8.574 \times 10^{-3}$	1.2	$1.339 \times 10^{-2}$	1.9
	Square	$7.825 \times 10^{-3}$	1.4	$1.253 \times 10^{-2}$	2.7

**Table 5.12a. Optimum performance of the linear and square law DWVMF for Figure 5.5b. for 10% and 20% impulse noise. The best value is shown in bold. For the linear case, the relevant parameter is  $\alpha$ , while for the square law case the relevant parameter is  $\beta$ .**

		30% noise		40% noise	
Window Size	Weighting Function	NMSE	$\alpha, \beta$	NMSE	$\alpha, \beta$
$3 \times 3$	Linear	$1.544 \times 10^{-2}$	6.8	$2.760 \times 10^{-2}$	8.7
	Square	$1.919 \times 10^{-2}$	11.1	$4.468 \times 10^{-2}$	14.2
$5 \times 5$	Linear	<b><math>1.409 \times 10^{-2}</math></b>	3.6	<b><math>2.071 \times 10^{-2}</math></b>	5.4
	Square	$1.751 \times 10^{-2}$	6.9	$2.858 \times 10^{-2}$	14.8
$7 \times 7$	Linear	$1.503 \times 10^{-2}$	3.2	$2.103 \times 10^{-2}$	4.6
	Square	$1.699 \times 10^{-2}$	6.6	$2.415 \times 10^{-2}$	17.1
$9 \times 9$	Linear	$1.784 \times 10^{-2}$	2.7	$2.380 \times 10^{-2}$	4.3
	Square	$1.909 \times 10^{-2}$	6.4	$2.601 \times 10^{-2}$	14.6

**Table 5.12b. Optimum performance of the linear and square law DWVMF for Figure 5.5b. for 30% and 40% impulse noise. The best value is shown in bold. For the linear case, the relevant parameter is  $\alpha$ , while for the square law case the relevant parameter is  $\beta$ .**



a



b



c



d



e



f

Figure 5.7a. Figure 5.5a contaminated with 30% impulse noise. Figure 5.7b. Application of standard VMF with window size  $3 \times 3$  to Figure 5.7a. Figure 5.7c–f. Application of DWVMF with window sizes  $3 \times 3$  to  $9 \times 9$  respectively with optimum values of  $\alpha$  to Figure 5.7a.





**a**



**b**



**b**



**c**



**d**



**e**

**Figure 5.8a.** Figure 5.5b contaminated with 30% impulse noise. **Figure 5.8b.** Application of standard VMF with window size  $3 \times 3$  to Figure 5.7a. **Figure 5.8c–f.** Application of DWVMF with window sizes  $3 \times 3$  to  $9 \times 9$  respectively with optimum values of  $\alpha$  to Figure 5.7a.

### 5.5.3 DISCUSSION

It seems relevant to recall that, in general, real-world sampled images exhibit a greater abundance of low-frequency information than high-frequency information. Indeed, the majority of modern day image compression algorithms exploit this tendency by attaching greater importance to low-frequency coefficients with respect to quantization and storage. High frequency information in images is usually due to:

- Object edges
- Complex textures
- Impulse noise

It seems clear from this list that algorithms used to minimise impulse noise may also destroy edge information and texture detail, due to their similarities in spatial frequency. It is well known that the noise removal efficacy of a sub-image window filter improves as the area of the operator increases (for a mean filter the signal to noise ratio increases as the root of the number of elements under consideration), though the simultaneous increase in area exacerbates the problems of blurring and edge-shifts.<sup>120</sup>

In the filter presented here, it is assumed that as low frequencies are predominant, the immediate neighbourhood surrounding a given pixel will have little variation or will be changing monotonically. Should the central pixel in the neighbourhood be noisy, the hypothesis is that a noise-free point physically close to the central pixel will be closer in value to the original, uncorrupted value than a point further away – with the exception of areas of homogeneous texture with a texel size significantly smaller than the neighbourhood under consideration.

However, as noted above, small neighbourhoods are poor at suppressing more than minor amounts of noise. Hence the idea of distance weighting is introduced; in this manner the advantages of a small neighbourhood are retained, together with the superior noise rejection capability of large neighbourhoods.

To demonstrate this, two examples of natural colour, real-world images (Figure 5.5) were examined; the first order statistics of varying size sub-image neighbourhoods at ten random points were analyzed, and the positions of the mean and median vectors, and their respective distances from the original, central pixel were noted.

Although the variance might be expected to increase as the neighbourhood enlarges (Figure 5.9), it should be noted that unless the variance increases isotropically the position of the mean and median will be affected (Figures 5.10 and 5.11). It can be seen that, whilst the results are highly nonlinear, the measurements show a general increase in the Euclidean distance between the median vector and the original pixel vector as the neighbourhood size increases. However, in some instances the distance is reduced; examining the actual pixel values here show these areas to be regions of characteristically similar texture, and as such the increase in neighbourhood size makes the distribution more symmetrical around that pixel, reducing its Euclidean distance to the calculated median vector.

Having shown by way of image statistics the motivation of using smaller neighbourhoods for obtaining a filtered image remaining close to the original, the results of the experiments bear out the hypothesis developed. As already noted, the optimal choice of weighting function clearly depends on the type of image under consideration and its statistics; the experiments show the linear law DWVMF to be appropriate (but possibly sub-optimal) for processing natural colour images, whereas the square law weighting function used in the second set of experiments is clearly inappropriate.

#### **5.5.4 CONCLUDING REMARKS**

A new filter based on weighted median filters which are already well-known has been introduced, and the rationale for a filter based on the spatial properties of local properties has been discussed. In addition, its use on natural colour test images has been demonstrated.

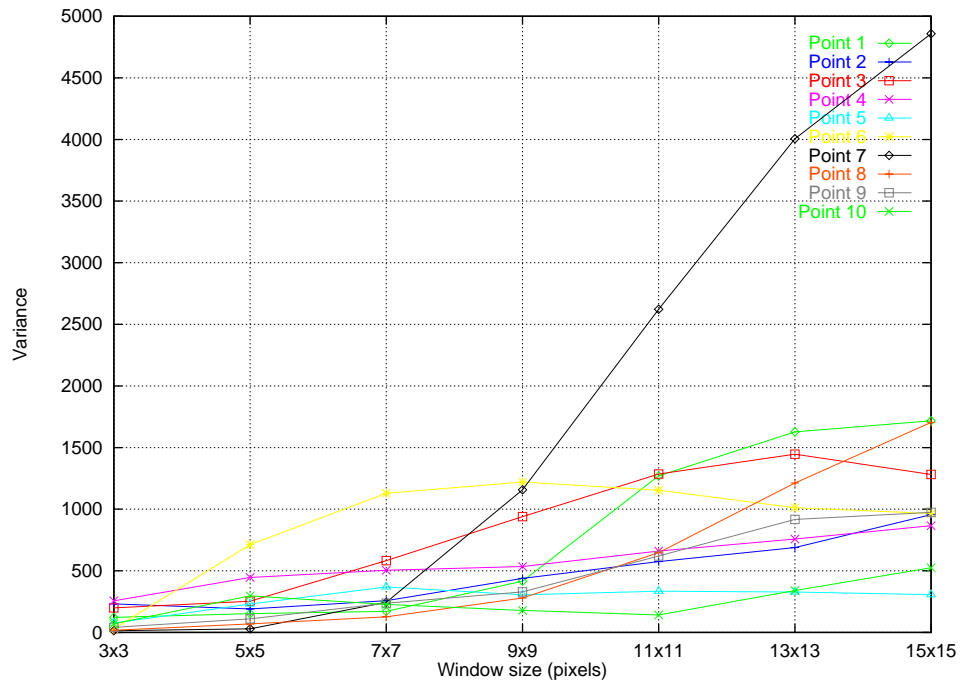


Figure 5.9a. Variance of test images for neighbourhood sizes  $3 \times 3$  to  $15 \times 15$  at ten random positions in Figure 5.7a.

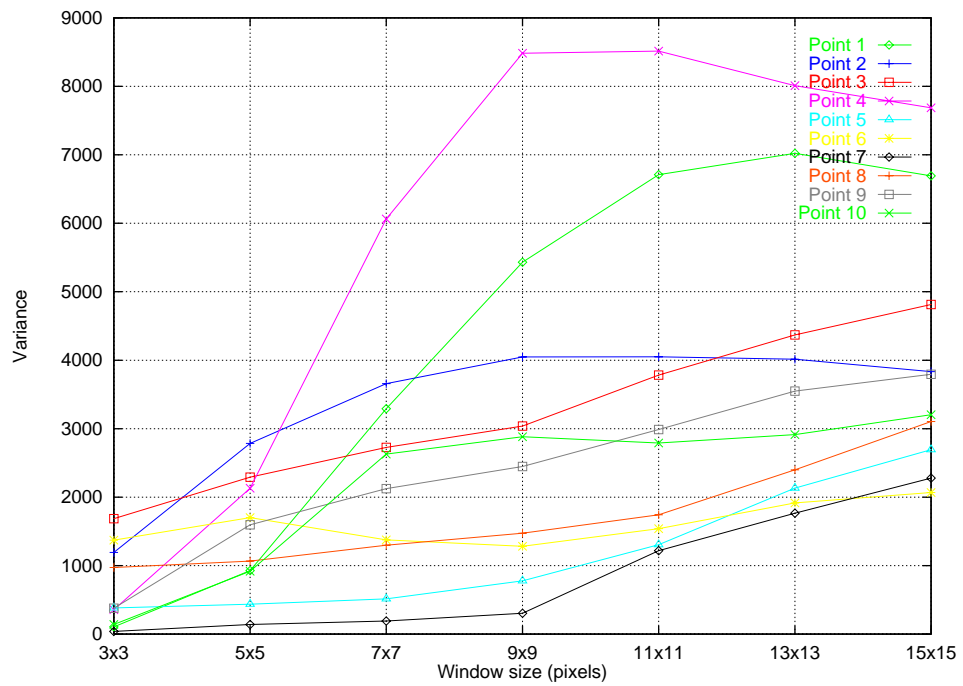


Figure 5.9b. Variance of test images for neighbourhood sizes  $3 \times 3$  to  $15 \times 15$  at ten random positions in Figure 5.7b.

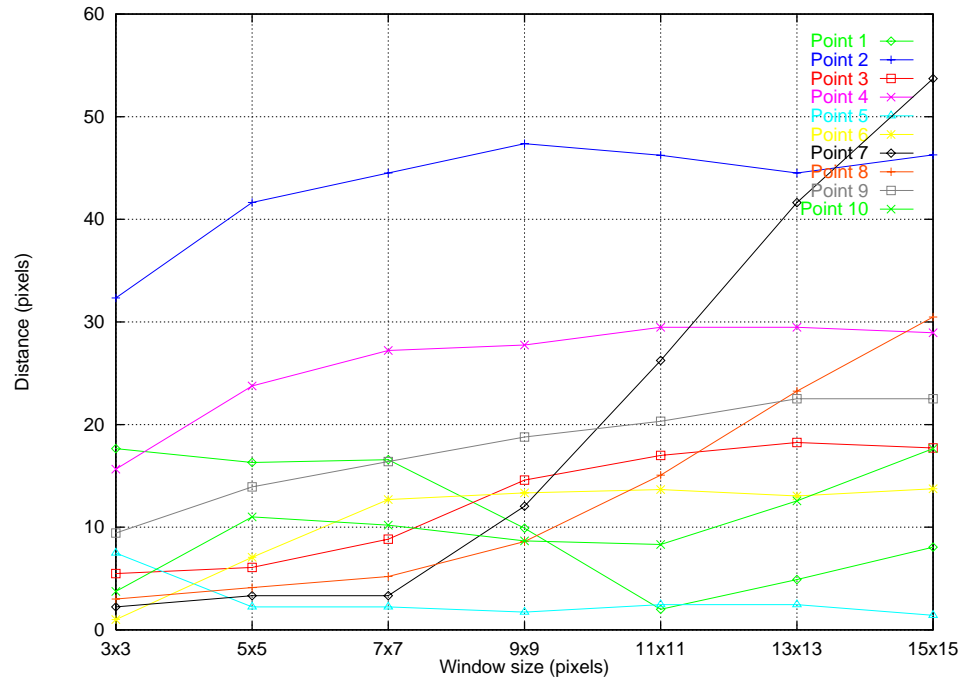


Figure 5.10a. Euclidean RGB distance of mean from original pixel in test images for neighbourhood sizes  $3 \times 3$  to  $15 \times 15$  at ten random positions in Figure 5.7a.

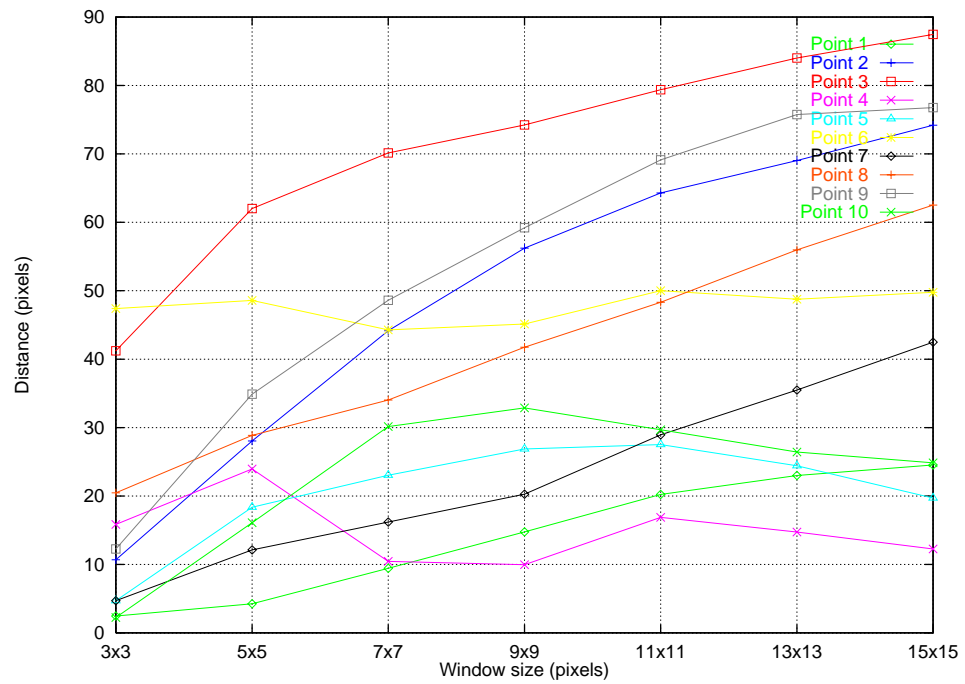


Figure 5.10b. Euclidean RGB distance of mean from original pixel in test images for neighbourhood sizes  $3 \times 3$  to  $15 \times 15$  at ten random positions in Figure 5.7b.

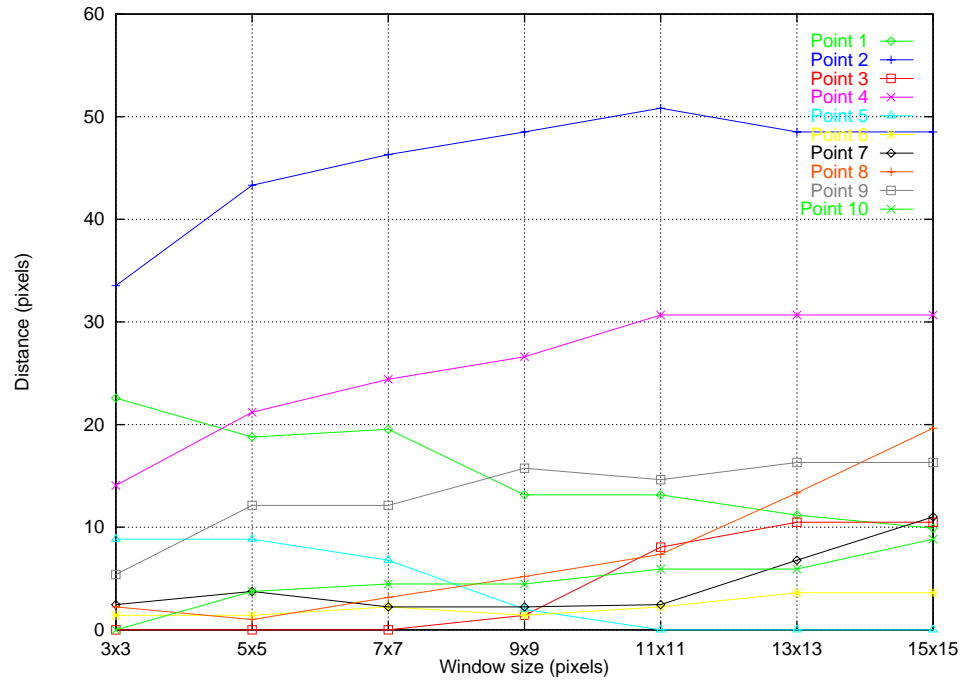


Figure 5.11a. Euclidean RGB distance of median from original pixel in test images for neighbourhood sizes  $3 \times 3$  to  $15 \times 15$  at ten random positions in Figure 5.7a.

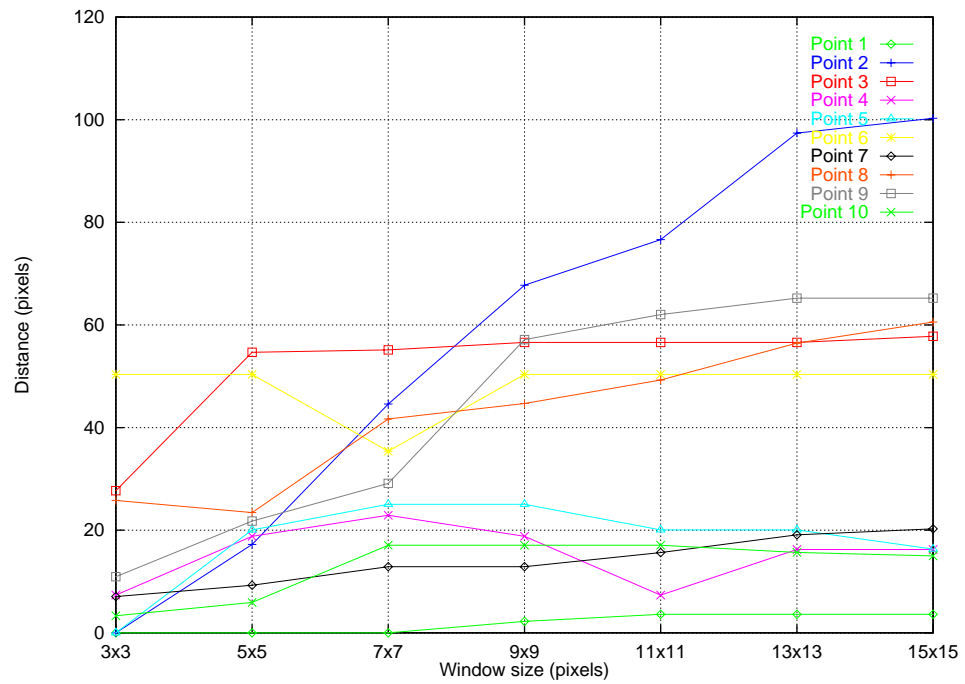


Figure 5.11b. Euclidean RGB distance of median from original pixel in test images for neighbourhood sizes  $3 \times 3$  to  $15 \times 15$  at ten random positions in Figure 5.7b.

The effectiveness against the well-known vector median filter has been shown and the relative trade-offs discussed. With simple adjustment of weighting function and given parameters, this filter may be utilised with known degree of efficacy against classes of image and noise models known *a priori*; as the optimum value of  $\alpha$  increases with the level of noise, a preset value may be used when the noise level is known. To improve the response of the filter, an adaptive weighting system may be employed in future work based on the analysis of local image statistics to reject noise at object boundaries more effectively, again with the goal of retaining as much of the original, uncorrupted image detail as possible. For instance, recent work by Eng and Ma<sup>153</sup> describes methodologies for contamination estimation and operator size switching; this may be utilized to calculate both the appropriate value of  $\alpha$  and the correct window size to use in a localized area.

## 5.6 FAST CALCULATION OF THE VMF

Although the rationale for preferring a vector method of obtaining the median of a multivariate dataset over component-wise application of the scalar median filter was discussed in Chapter 4, it is clear that the vector median filter is far more computationally expensive than the scalar median filter.

When employing the  $L_2$ -norm as the distance measure for a sample set consisting of  $M$ -dimensional vectors,  $d_{ij}$  is calculated as:

$$\sqrt{\sum_{k=1}^M (i_k - j_k)^2} \quad (5.11)$$

For a sample set containing  $N$  vectors (discounting  $i = j$  pixel pairs), a basic implementation of the VMF requires  $N(N-1)$  distances to be evaluated. Clearly, this can be a time-consuming task for large  $N$ , approximating to  $O(N^2)$  – particularly as the square root operation is relatively slow on many processors. Clearly, any method which minimizes processing time will be welcome. In fact, fast versions of the vector median filter have already been developed by Barni et al by approximating

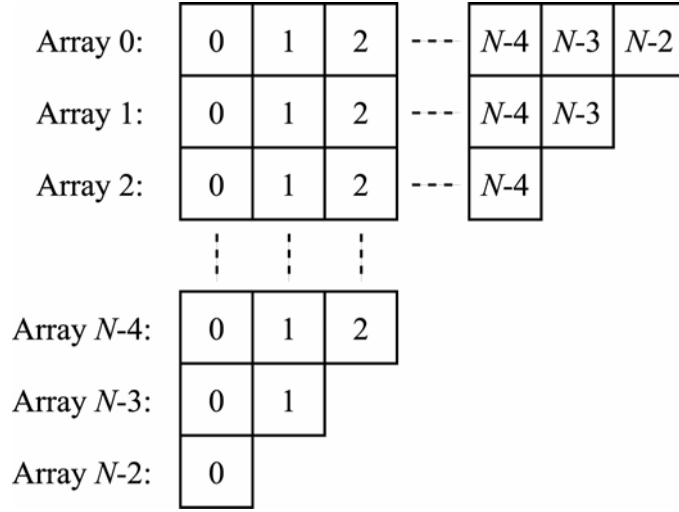


Figure 5.12. Array allocation for vector distance look-up table.

the ‘city-block’ ( $L_1$ ) and Euclidean-squared norms. A first approximation of the  $L_1$  norm filter is achieved by component-wise application of the scalar median filter resulting in an output vector  $\mathbf{x}_a$ . The vector in the dataset which minimizes the distance to  $\mathbf{x}_a$  then approximates the output of the  $L_1$ -norm VMF.<sup>81</sup> This fast method can be improved to properly obtain the  $L_1$ -norm by integrating along the path between  $\mathbf{x}_a$  and the true  $L_1$  median.<sup>123</sup>

The fast Euclidean-squared norm is obtained by finding the vector which minimizes the distance to the centroid of the dataset, i.e. the arithmetic mean.<sup>81</sup> This has also been used in conventional scalar image filtering to good effect.<sup>80</sup>

No fast algorithm for true calculation of  $L_2$ -norm vector median operator has been proposed so far, only a fast method which approximates the Euclidean distance between vectors.<sup>124</sup>

The fast method presented here requires only  $0.5N(N-1)$  operations – half the amount of computation of calculating each pixel pair distance, and it should be noted that it does not rely on making approximations. However, storage for a look-up table of  $0.5N(N-1)$  elements is required. This method works by exploiting a redundancy in distance calculations, namely that  $d_{ij} = d_{ji}$ .



Firstly, memory is allocated for  $N-1$  arrays; the first array contains  $N-1$  elements, the second  $N-2$ , and so on, as shown in Figure 5.12.

The  $i$ th element in the  $j$ th array contains the Euclidean RGB distance  $d_{ij}$ . As  $d_{ij} = d_{ji}$ , the way in which the look-up table is accessed is quite simple –  $\min(i, j)$  provides the array to be accessed, and  $\max(i, j)$  gives the offset into that array.

The table layout corresponds to the number of calculations required to complete the sum-of-distances for each successive sample; i.e. when the last sample is visited all but one of the inter-sample distances have already been calculated, and can be obtained from a fast access into the look-up table.

This method of using a look-up table is especially useful when one realizes that a  $w \times w$  square sub-image window only admits  $w$  new samples when it is shifted one element across – with this method only  $w(N-w) + 0.5w(w-1)$  new distances must be calculated, as the rest are already present in the look-up table. Such a ‘running median’ filter is discussed in Pitas, 1993.<sup>125</sup> The use of octagonal windows to implement running medians has also been discussed.<sup>70</sup>

### 5.6.1 IMPLEMENTATION

The introduced technique was implemented on an Intel Pentium III-class PC running at 700MHz under Windows 98, and tested on two 24-bit images, of size  $128 \times 128$  and  $256 \times 256$ . Initial tests showed little or no improvement over the standard  $L_2$ -norm VMF; it was deduced that this null result stemmed from memory page faults incurred by non-linear access to the memory, causing stalls and on-processor cache RAM misses. A comparative table of operation and access timings is given in McConnell, 1993.<sup>126</sup>

The method was recoded to use a contiguous section of memory, with a revised access scheme. A single block of memory of size  $0.5N(N-1)$  was allocated, and again as  $d_{ij} = d_{ji}$ , the way in which the array is accessed is quite simple, using the following relation:

```

if (i != j)
{
    if (i > j)
    {
        m = j;
        o = (N-i);
    }
    else
    {
        m = i;
        o = (N-j);
    }
    e = (m*N - 0.5*(m*(m-1))) + o - 1;
}

```

**Table 5.13. C language code to select correct array element.**

$$e = \left( mN - \frac{m(m+1)}{2} \right) + o - 1 \quad \begin{cases} m = j, & o = (N-i) & i > j \\ m = i, & o = (N-j) & i < j \end{cases} \quad (5.12)$$

where  $e$  is the array element to be accessed to give  $d_{ij}$ , assuming the array index starts at zero. The simple C language equivalent is given in the section of code in Table 5.13.

## 5.6.2 RESULTS AND DISCUSSION

Once recoded in this manner, significant reduction in computation was observed, shown in Table 5.14.

It is apparent from the table that time savings of approximately 30% are achievable, but not the 50% suggested by the algorithm initially. This is due to the overhead involved in managing the memory allocation and performing the array access.

Naturally, this technique may also be used for other  $L_p$ -norm metrics, although in general only the  $L_1$ ,  $L_2$  and  $L_\infty$  norms are employed.<sup>28</sup>

Window size ( $N$ )	128x128, time in seconds		256x256, time in seconds	
	Original VMF	Optimized VMF	Original VMF	Optimized VMF
3x3 (9)	0.22	0.16	0.93	0.61
5x5 (25)	1.48	0.99	6.26	4.01
7x7 (49)	6.28	3.35	26.42	14.28
9x9 (81)	16.48	10.16	70.58	43.94
11x11 (121)	30.64	20.15	132.31	86.68
13x13 (169)	58.88	38.11	260.62	170.05
15x15 (225)	100.68	66.88	452.42	303.74

**Table 5.14. Computation time for original and optimised vector filters, carried out on AMD Athlon 700MHz with 256Mb RAM running Windows 98.**

As already noted, the memory management takes up a significant proportion of the algorithm's time, and as the computational burden is not as great for the  $L_1$  and  $L_\infty$  norms as for the  $L_2$ -norm the performance advantage of the new method presented here is somewhat lessened and may be negated completely in some systems due to the time needed for the array set-up and the subsequent look-up.

## 5.7 CONCLUDING REMARKS

This chapter has introduced a number of new results for and extensions to the vector median filter, namely:

- The measurement of the efficacy of the standard VMF in alternative colourspaces.
- A description and implementation of a hybrid mean-median vector filter as an alternative to the  $\alpha$ VMF.

- A description and implementation of a rank-order filter with a hard decision boundary to preserve details while removing noise.
- A weighting regime based on the statistics of natural images to improve detail-preservation while removing impulse noise.
- An algorithmic adaptation of the distance calculations necessary for rank order filtering to improve computational efficiency.

These extensions have been discussed and compared with the standard VMF in a context that may be compared with the filters discussed in the last chapter, and the improvements gained in performance are clear.

# Chapter 6: Mode Filtering

## 6.1 INTRODUCTION TO MODE FILTERING AND THE TRUNCATED VECTOR MEDIAN FILTER

It was discussed in Chapter 4 that the median sample maximizes the likelihood function in a double exponential input distribution, and is well suited for impulse noise removal.<sup>74</sup> Similarly, for a symmetric Gaussian distribution, it is known that the mean of the distribution is the optimal estimate of the maximum likelihood location parameter.<sup>80</sup>

However, for data sets whose distributions are not symmetric Gaussian or exponential, it is evident that the mode should be selected, as this represents the most probable value. In spite of this the mode filter is used relatively rarely; notable exceptions are Coleman and Andrews (1979),<sup>132</sup> and Evans and Nixon (1995).<sup>133</sup> This rare incidence of use must be partly because of implementation difficulties. Indeed, it was pointed out some time ago that, while it is trivial to find the highest point in the intensity histogram, it is far from easy to find the underlying mode in an intensity histogram.<sup>134</sup> Furthermore, the specification problem is less straightforward, and this must also contribute substantially to the relative lack of application of this type of filter. These facts are evidenced by the paucity of papers on the subject.

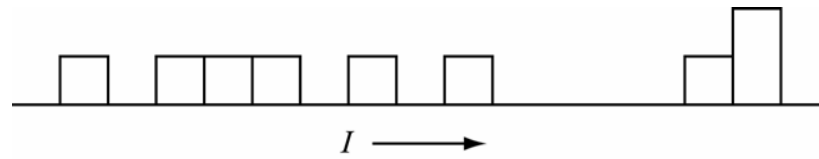
In spite of these difficulties, the mode filter was developed in a form known as the ‘truncated median filter’, and was found to be highly effective for enhancing grey-scale images.<sup>134</sup> Since then the method has been applied to the accurate, robust location of Hough transform peaks.<sup>135</sup> In addition, some of its properties have been studied in more detail. In particular, analysis has shown that the mode filter acts in the same way as the median filter in shifting curved edges, so in this respect the mode filter appears to offer no special advantage.<sup>119,136</sup>

This chapter is aimed at further understanding of the operation of the mode filter, and particularly at finding whether it can be adapted to enhance colour images. In principle there could be problems in achieving this, as histogram-based processing methods can run into trouble if they involve ordering. This is because colour spaces are 3-D rather than 1-D, and thus ordering is, *a priori*, a meaningless concept. However, it is shown that the 1-D histogram processing method adopted by Davies<sup>134</sup> can be extended to permit mode filtering based on the truncated median filter concept. In fact the performance of the technique is shown to be impressive, not only for image enhancement but also for noise elimination.

Section 6.2 outlines the problems involved in mode filtering, and shows how the truncated median filter operates on a standard grey-scale intensity histogram. Section 6.3 develops the methodology for applying the technique to colour images. Section 6.4 describes its performance on colour images and offers various insights into its effectiveness both for enhancement and for noise elimination. Section 6.5 examines the edge shifting characteristics of the mode filter, and discusses the significance of the results. Section 6.6 presents a finite edge width model which leads to numerical estimates of breakpoints between the various regions on the edge shift graphs. Section 6.7 summarises the situation and gives further motivation for the use of mode filters.

## **6.2 PROBLEMS OF MODE FILTERING AND A SOLUTION**

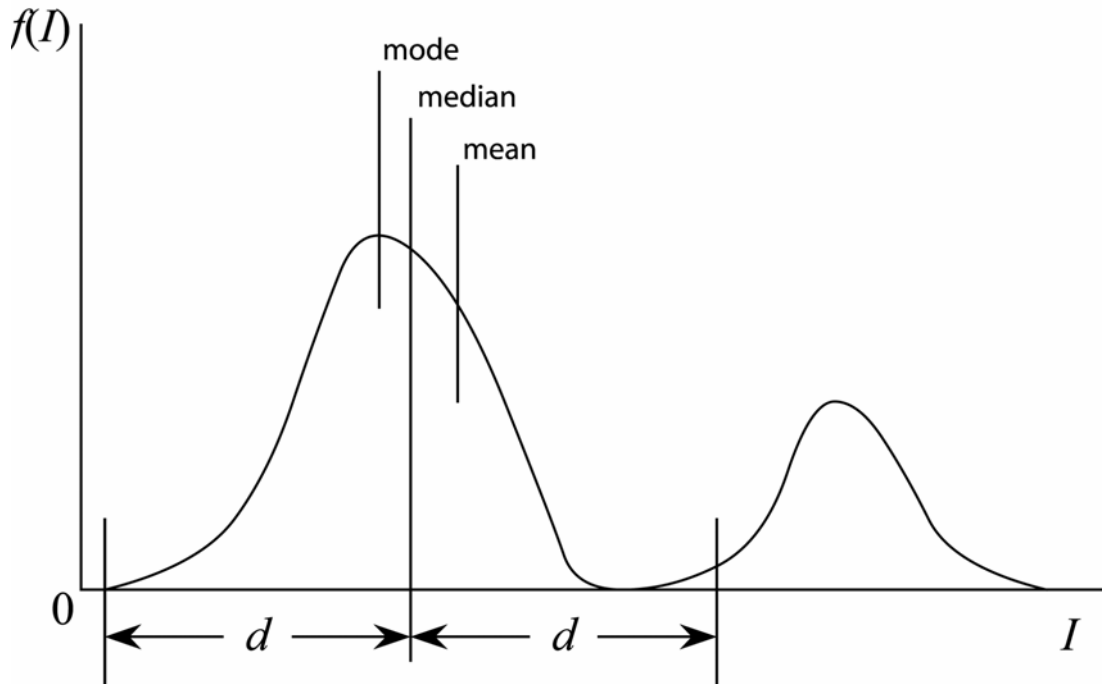
Before considering the problems involved in designing and implementing a mode filter, it is worth considering briefly the properties it is likely to have. By definition, the mode filter will take the intensity distribution in any neighbourhood and will return the most probable value corresponding to the highest point in the distribution. At a location in the middle of a plain (non-textured) object the main effect will be to eliminate noise: indeed, if the noise is taken to be Gaussian, and thus to have a symmetrical intensity distribution, the mean, median and mode will be coincident, and all these types of filter will give similar responses. The same will happen in a plain background region. On the other hand, very near the edge of an object, the



**Figure 6.1.** Example of difference between the highest point and the underlying mode in the sparse local intensity histogram of a  $3 \times 3$  neighbourhood.

intensity will be slowly varying, and the mode filter will take a majority decision on the intensity value, with the result that it will tend to maintain the object intensity right up to the edge. Similarly, in the background region, the background intensity will be maintained right up to the edge. This means that the transition from dark to light will become significantly more sudden and pronounced. This makes it clear that the mode filter will have a strong effect on edges, ‘crispens’ the image and enhancing contrast, while well away from edges, the effect will be mainly to eliminate noise. Note that the performance in the presence of impulse noise should be better than that for a mean filter (in the sense that the noise will be ignored rather than averaged in): in this respect the performance should approach that attained by a median filter.

Unfortunately, a tedious problem arises as soon as we attempt to apply this idea. The local intensity distribution is bound to be rather sparsely populated for the window sizes normally used for image filtering. As a result, it will generally be multi-modal and far from smooth. Hence the highest point may well be at a rather arbitrary location and may not indicate the position of the *underlying* mode (Figure 6.1). This means that the distribution will need to be smoothed out considerably before the mode can be computed. However, the width of the distribution may vary widely from region to region (e.g. from close to zero to close to 256), so it will be difficult to specify the amount of smoothing needed in any instance. This means that it will be better to choose an indirect measure of the position of the mode rather than attempt to measure it directly. These caveats preclude the use of a density gradient estimator,<sup>139,140</sup> which as well as being computationally intensive relies on significantly sized sample sets to obtain the mode; for instance this technique has recently been used to segment colour images.<sup>141</sup>



**Figure 6.2. Method of truncating the local distribution.** The diagram shows the ordering of the three means for a bimodal distribution. It also shows how the intensity histogram is truncated at a specified distance from the median. The rationale for this procedure is indicated by the idealised situation which would exist if the position of the mode were known initially.

An alternative method of tackling these problems is to estimate the position of the mode from various other statistics of the local intensity distribution. In particular, it is widely known for large smoothly varying distributions that the separations between the mean, median and mode are related by the formula:<sup>134</sup>

$$\text{mean} - \text{mode} \approx 3.0(\text{mean} - \text{median}) \quad (6.1)$$

However, it has been shown that for the bimodal distributions that arise at the edges of objects, this formula gives erroneous estimates of the mode position and is not practically useful.<sup>134</sup> This implies that a special strategy will have to be adopted for coping with bimodal distributions in this sort of application. Specifically, the concept is to eliminate the smaller peak, thus leaving the large peak intact, so that its exact position can be estimated accurately without interference. To achieve this, it was noted by Davies that the median provides a robust estimate somewhere between the larger and smaller peaks, though closer to the larger peak position.<sup>134</sup>



```

repeat    // as many passes over image as necessary
  repeat    // for each pixel
    compute local intensity distribution;
    repeat    // iterate to improve estimate of mode
      find minimum, median and maximum intensity values;
      decide from which end local intensity distribution should be truncated;
      deduce where local intensity distribution should be truncated;
      truncate local intensity distribution;
      find median of truncated local intensity distribution;
    until median sufficiently close to mode of local distribution;
    transfer estimate of mode to output image space;
  until all pixels processed;
until sufficient enhancement of image;

```

*Comments :*

- (i) The outermost and innermost loops can normally be omitted (i.e. they need to be executed once only).
- (ii) The *final* estimate of the position of the mode can be performed by simple averaging instead of computing the median: this has been found to save computation with negligible loss of accuracy.
- (iii) Instead of the minimum and maximum intensity values, the positions of the outermost octiles (for example) may be used to give more stable estimates of the extremes of the local intensity distribution.

**Table 6.1. Truncated median filter algorithm.**

Thus measuring from the median to the closer end of the distribution, and then moving an equal distance to the other side of the median provides a useful indicator of that portion of the distribution which can be eliminated (Figure 6.2). Repetition of this procedure will progressively eliminate the smaller peak and leave the larger peak intact, so that any convenient averaging process can be used to provide an estimate of the mode position. In practice, however, it was found unnecessary to repeat the process – substantial edge enhancement was achievable with a single

truncation. The result was called the ‘truncated median filter’ (Table 6.1).<sup>134</sup> The theoretical justification for this procedure is that if the position of the mode were known, the small peak elimination procedure would be essentially exact for a bimodal distribution. However, as the median is known to lie further from the mode (the main peak) it is somewhat less effective at removing the minor peak (Figure 6.2): i.e. this is a ‘cautious’ algorithm which errs on the side of safety: it cannot eliminate too much of the original distribution and thereby destroy necessary information.

The effectiveness of this algorithm was amply demonstrated for grey-scale images<sup>134</sup> (examples of this will be shown below). In what follows the algorithm is extended for application to colour images.

### 6.3 METHODOLOGY FOR APPLYING MODE FILTERING TO COLOUR IMAGES

To extend the truncated median filter concept to colour, it must be made applicable in the 3-D colour space. In fact, by considering a distribution in 2-D, it is shown how to extend the filter to the 3-D case: it will also be important to confirm that the new algorithm gives identical results to the scalar version when reverting to 1-D.

First, note that the distribution of pixel intensities in 1-D is given as an intensity histogram or ordered list of values, and the median is calculated as the value at the middle of this range. However, as remarked in Section 4.4, the ordering concept is meaningless in a multi-dimensional space, and instead the property of the median is that it is the vector which minimises the sum of the absolute distances  $|d_{ij}|$  to the other data points must be used to generalise to higher dimensional situations.

$$median = \min_i \sum_j |d_{ij}| \quad (6.2)$$

Or, repeating Equation 4.17:

$$\mathbf{x}_{med} = \min \sum_{i=1}^N |\mathbf{x}_i - \mathbf{x}_j| \quad \forall \mathbf{x}_j$$

(For convenience, the result of this type of calculation shall be continued to be called the median.)

It can be seen in the 2-D case (Figure 6.3a) that although the median  $M_1$  lies close to the centroid of those input vectors which are normally distributed, the presence of outliers skews the overall distribution, and neither the median nor the mode is well approximated by the centroid. As in the 1-D case (Section 6.2), we seek to address this by removing the outliers from the calculation: this will have to be achieved by truncating the input vector distribution appropriately.

In the sample distribution shown in Figure 6.3a, contaminated by two outliers, the enclosing ellipsoid is egg-shaped, with the centroid located nearer the broad end. However, if the outliers are removed the distribution becomes normally distributed with an approximately uniform circular deviation (Figure 6.3b). In that case the centroid will move towards the broad end of the distribution – approximately in the direction of the vector  $N_2 \rightarrow M_1$ .

Since the position of the mode is not known *a priori*, the truncation must be performed in a safe manner, i.e. care must be taken not to truncate too much of the distribution but still move the centroid closer to the underlying mode. The technique that we have designed to accomplish this in multi-dimensional cases works as follows:

1. Determine the median  $M_1$  of the input distribution according to Equation 6.2.
2. Find the furthest outlier  $O$  of the distribution, i.e. the input vector which maximises the sum-of-distances cost function (this is also used for certain noise metrics<sup>91</sup>).
3. Calculate the vector  $\mathbf{v}$  from  $M_1$  to  $O$  and move from  $M_1$  through a distance  $-\mathbf{v}$ . Mark this position as reference point  $P$  (Figure 6.3c).  $P$  will now be outside the input distribution, and will also be further away from the most distant outlier than any of the other inputs.

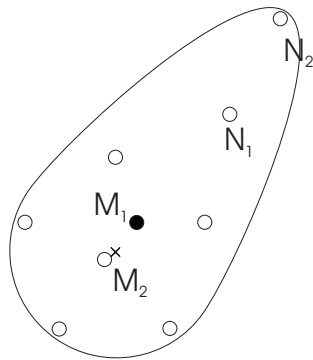


Figure 6.3a

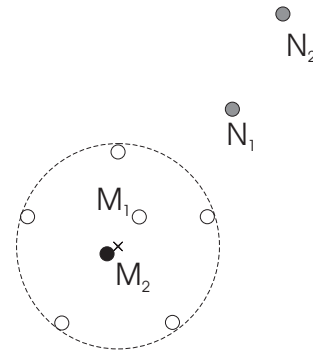


Figure 6.3b

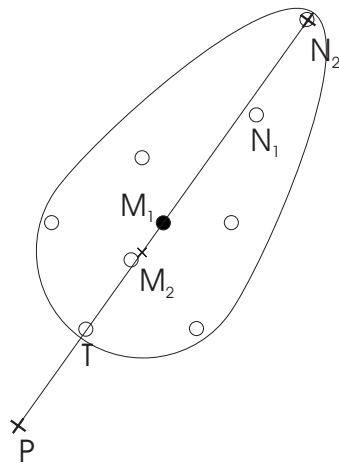


Figure 6.3c

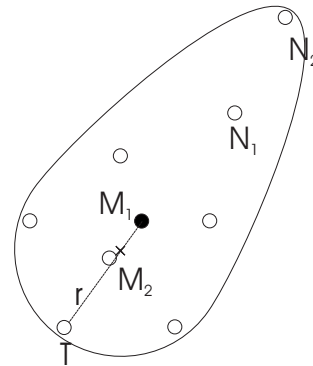


Figure 6.3d

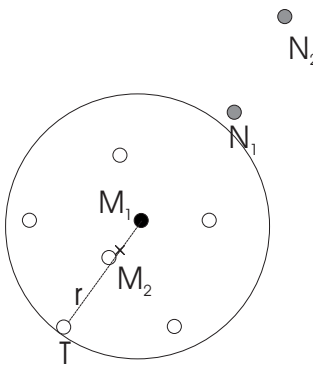


Figure 6.3e

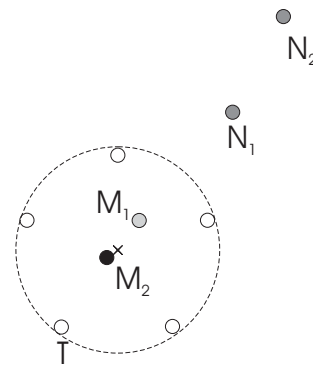


Figure 6.3f

**Figure 6.3.** Method of truncating the local distribution in 2 dimensions, extendible to higher dimensional spaces. (a) shows complete input distribution with median vector  $M_1$  selected by VMF and ideal median  $M_2$ . (b) shows the truncated input distribution with ideal median  $M_2$  selected. (c)–(e) demonstrate the method used to truncate the local distribution. Noise vectors are removed from consideration by locating a point  $P$  equidistant and diametrically opposite to the median from the furthest outliers. The truncation occurs at a radius from the median determined by the input vector closest to  $P$ . (f) shows the new local distribution from which a more accurate estimate of the centroid is calculated.

4. Find the input vector  $T$  closest to  $P$  (Figure 6.3c).
5. Determine the distance  $r$  from  $M_1$  to  $T$  (Figure 6.3d).
6. Truncate the distribution at distance  $r$  from  $M_1$  (Figure 6.3e).
7. Find the median  $M_2$  of the new distribution (Figure 6.3f).
8. If  $M_1 = M_2$ , terminate the procedure (i.e. root signal has been found and no further truncation is possible): otherwise go to step 2.

In step 7, if  $O$  is the only outlier, this will now be removed from consideration and the ideal centroid position can be calculated. If any other outliers lie at a significant distance from the original centroid, these will also be removed. If, however, there is an outlier on the side of the distribution opposite to  $O$ , this will probably be selected as the input vector closest to  $P$ . In this case the filter behaviour will be sub-optimal because the input distribution will not be truncated sufficiently, although the centroid will have been moved towards the mode. Thus, like the scalar truncated median filter, this filter falls in to the category of a ‘cautious’ algorithm (see Section 6.2). It can be seen from the example distribution in Figure 6.3 that repeated application of the new filter to the truncated data set does not merely result in a new estimate for the median position. Whilst the above procedure may seem computationally intensive, it has already been noted by Davies that one pass of such a filter is sufficient for most applications.<sup>134</sup>

Finally, it should be noted that if more than one pass is conducted and the resulting distribution has an even population, the operator works in a similarly ‘safe’ manner as discussed in Section 6.2. Normally if a distribution has an even population, the average of the two median values is taken, and this would introduce a vector that is not part of the initial distribution. For this reason, when two median values arise in the present application, the one that is closest to the old median is selected. Clearly this applies to both scalar and vector versions of the truncated median filter.

## 6.4 APPLICATION OF MODE FILTERS TO COLOUR IMAGES

### 6.4.1 QUALITATIVE RESULTS

To demonstrate the characteristics of the new filter, three original (uncompressed) digitised images were taken: these were selected to give a variety of spatial and chromatic characteristics (Figures 6.4a–c). Figures 6.4d–f show the  $5 \times 5$  median filtered images and Figures 6.4g–i show the  $5 \times 5$  truncated vector-median (‘mode’) filtered images (TVMF).

The mode-filtered examples are seen to give objects a more well-defined edge outline, in line with the findings of Davies.<sup>134</sup> However, fine detail is lost within object boundaries, as one would expect from a filter selecting the mode of a distribution. It should be noted that the median-filtered images experience a similar loss of detail (this loss of detail is sometimes described as ‘softening’ to distinguish it from the blurring characteristic of mean filters).

For comparison, Figures 6.5a–c show the effects of component-wise filtering using the 1-D truncated median filter (TMF). The results show virtually identical characteristics to those for the truncated vector median filtered images, but with significant haloes and ‘colour bleeding’ around some object boundaries as a result of the component-wise filtering.

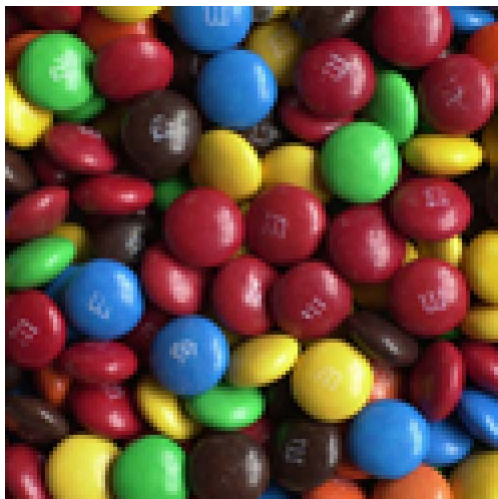
Colour bleeding is by now well known<sup>74</sup> and takes the form of edge-jitter which occurs when the median values from the processed channels do not correspond to a member of the input distribution when they are recombined to produce an output colour vector. This may happen when only one or two channels contain outliers, or when the input vectors show distinct clustering – as sometimes happens in the presence of noise or edges. Clearly, the effect is more noticeable the further the resultant vector is from the closest data point in the original distribution.



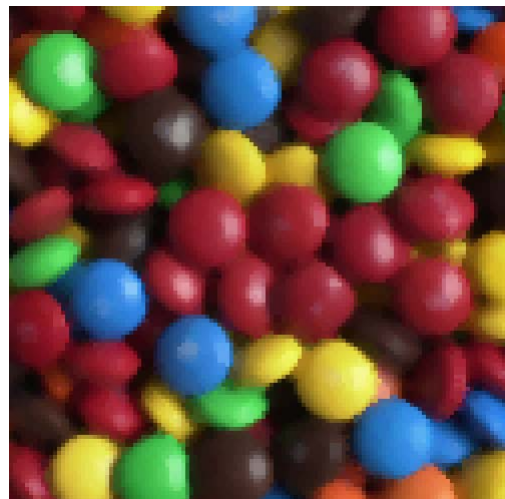
**Figure 6.4a.** Original image ( $128 \times 128$ ).



**Figure 6.4d.** Figure 6.4a after  $5 \times 5$  VMF.



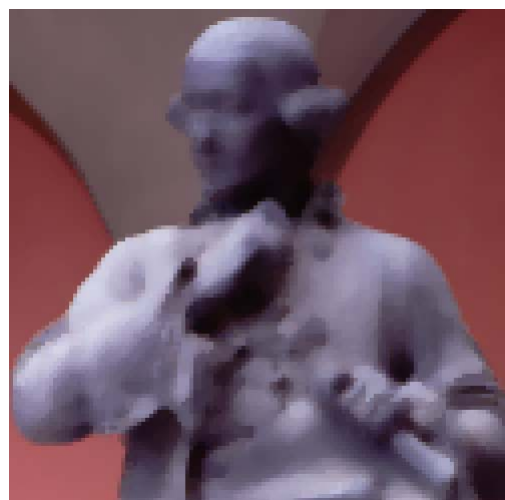
**Figure 6.4b.** Original image ( $128 \times 128$ ).



**Figure 6.4e.** Figure 6.4b after  $5 \times 5$  VMF.



**Figure 6.4c.** Original image ( $128 \times 128$ ).



**Figure 6.4f.** Figure 6.4c after  $5 \times 5$  VMF.

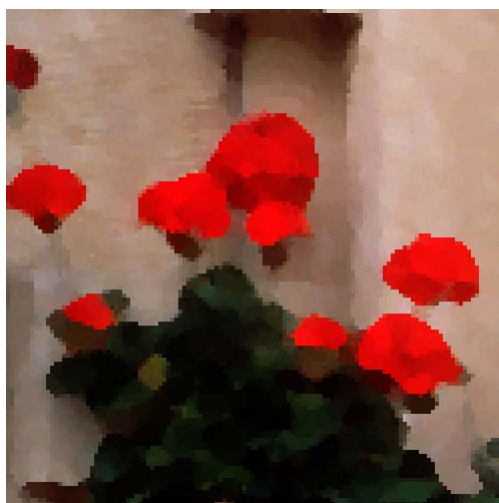


Figure 6.4g. Figure 6.4a after  $5 \times 5$  TVMF.

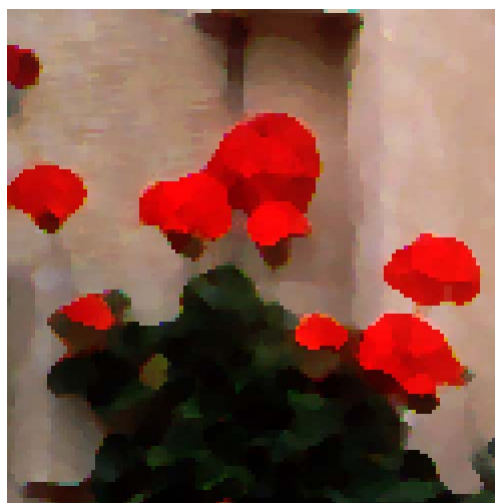


Figure 6.5a. Figure 6.4a after  $5 \times 5$  TMF.

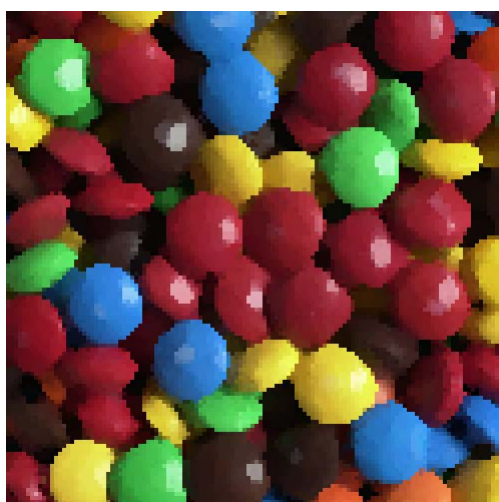


Figure 6.4h. Figure 6.4b after  $5 \times 5$  TVMF.

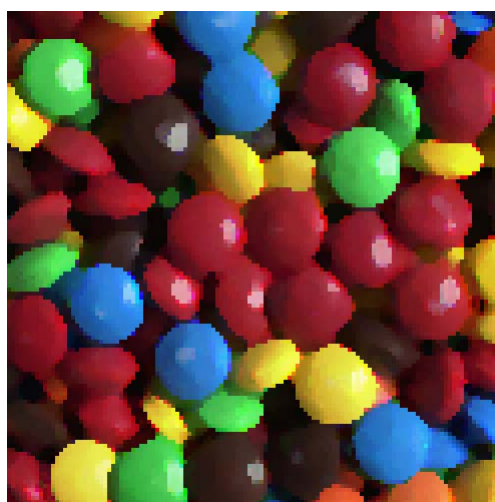


Figure 6.5b. Figure 6.4b after  $5 \times 5$  TMF.

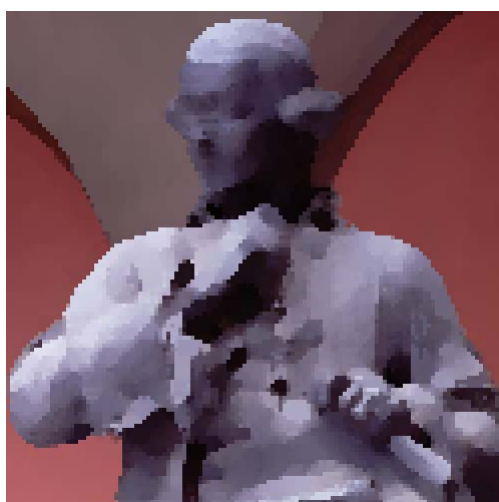


Figure 6.4i Figure 6.4c after  $5 \times 5$  TVMF.

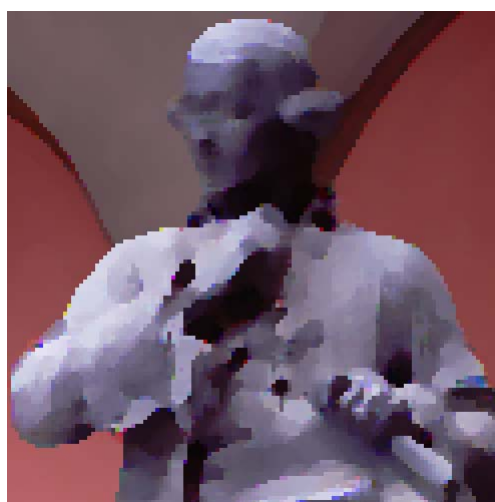


Figure 6.5c Figure 6.4c after  $5 \times 5$  TMF.



### 6.4.2 EDGE ENHANCEMENT

The relative edge strengths of the median and mode filtered images were measured using a Sobel  $3 \times 3$  edge-detection operator on each of the three colour channels. Figures 6.6a–c and Figures 6.7a–c show the Sobel edge components of the median- and mode-filtered images for the original images of Figures 6.4a–c: edge definition can be seen to be significantly stronger for the mode-filtered images.

By tabulating the pixel edge responses into ten 10% ranges and plotting the results graphically (Figures 6.8a–c), it can be seen that the mode filter does not tend to produce significant edges in the lower percentile ranges, but gives stronger responses than the median filter on object boundaries: this is indicated by the increase in intensity at the upper end of the percentile range.

### 6.4.3 LOCATION OF MODE

To demonstrate that the new filter does indeed perform well in estimating the mode of a given discrete distribution, note first that the mode and the mean usually lie on opposite sides of the median for a skewed distribution, and thus in general the Euclidean distance between the mode and the mean will be greater than that between the median and the mean.

A  $5 \times 5$  mean-filtered version of each original image was compared with the median- and mode-filtered images to test this hypothesis (Table 6.2), and it was found that in three cases, representing a variety of types of image, the above statement was valid (the respective numbers of pixels for which it applied were 91.7%, 88.1% and 86.6% of the image pixels). Of course, some distributions may be multi-modal, or may be contaminated with noise, or may appear random (a) because of the small number of elements within them and (b) because of discretely sampling the data. Clearly, in these cases the ordering of the means will not be normal,<sup>134</sup> and this seems to account for the small proportion of the pixels in the example images for which the hypothesis is found to be invalid.

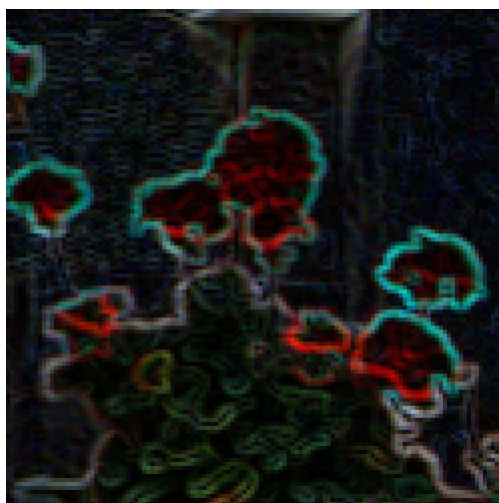


Figure 6.6a. Figure 6.4d after  $3 \times 3$  Sobel.

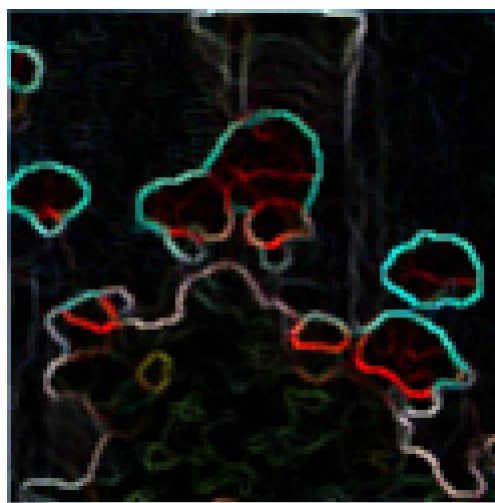


Figure 6.7a. Figure 6.4g after  $3 \times 3$  Sobel.

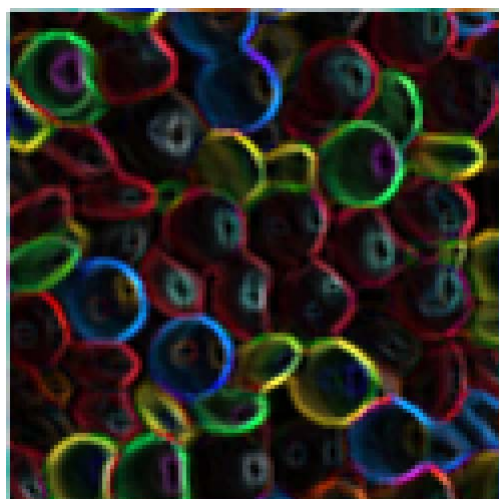


Figure 6.6b. Figure 6.4e after  $3 \times 3$  Sobel.

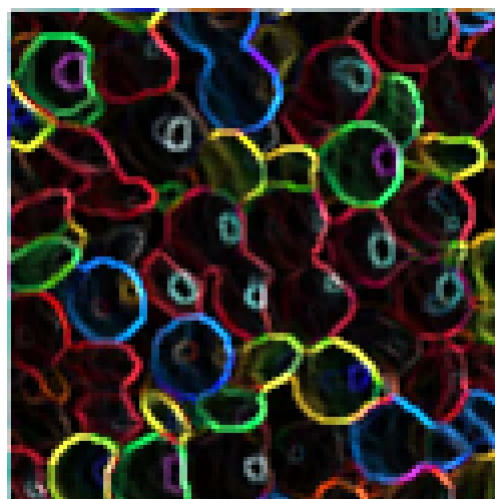


Figure 6.7b. Figure 6.4h after  $3 \times 3$  Sobel.



Figure 6.6c. Figure 6.4f after  $3 \times 3$  Sobel.



Figure 6.7c. Figure 6.4i after  $3 \times 3$  Sobel.

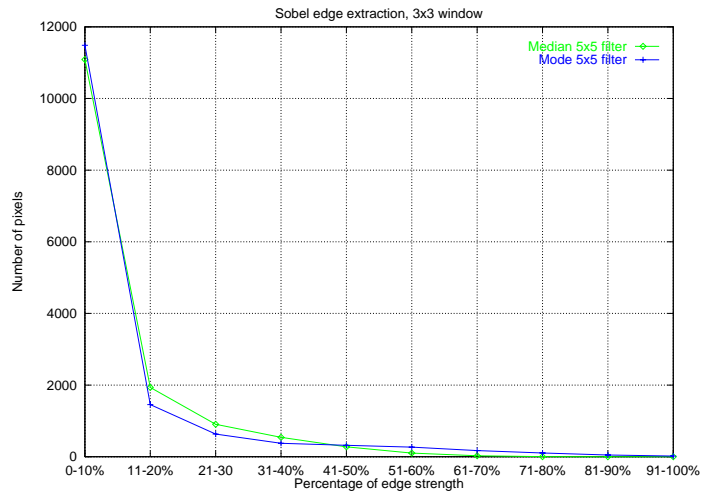


Figure 6.8a. Relative edge strengths for Figures 6.6a and 6.7a respectively.

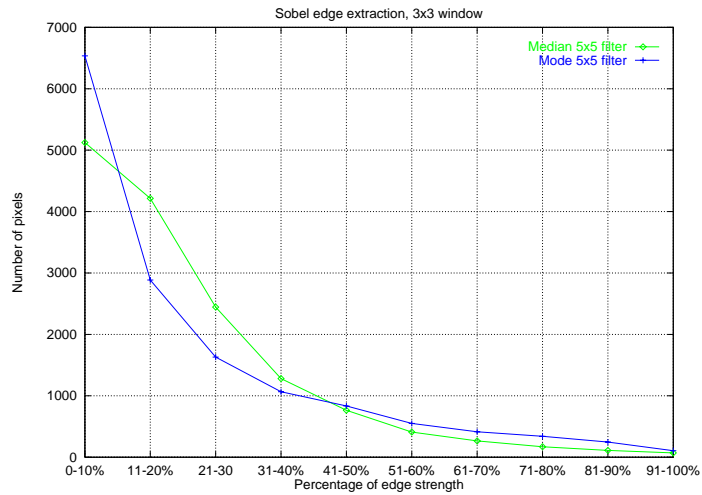


Figure 6.8b. Relative edge strengths for Figures 6.6b and 6.7b respectively.

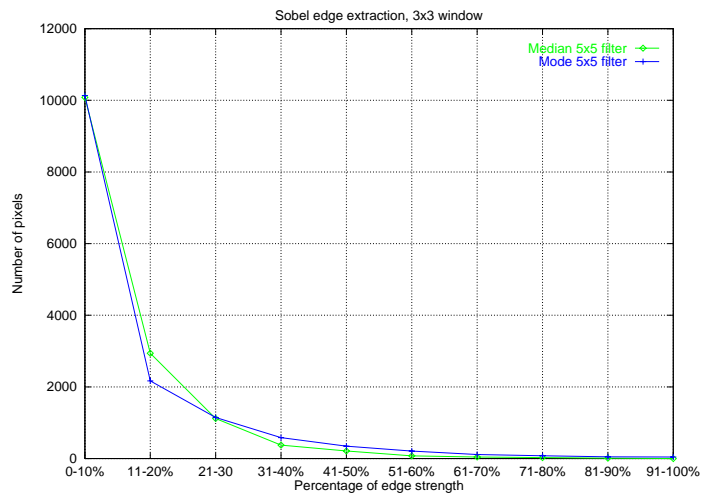


Figure 6.8c. Relative edge strengths for Figures 6.6c and 6.7c respectively.

Total number of pixels scrutinised (ignoring edge pixels due to $5 \times 5$ windows)	15376
No. of pixels for which Mode $\rightarrow$ Mean longer than Median $\rightarrow$ Mean	14096
Percentage	91.68%
No. of pixels for which Mode classified on opposite of plane normal to Median $\rightarrow$ Mean	8382
Percentage	54.51%

**Table 6.2a. Pixel classification statistics for ‘Flowers’ image (original in Figure 6.4a).**

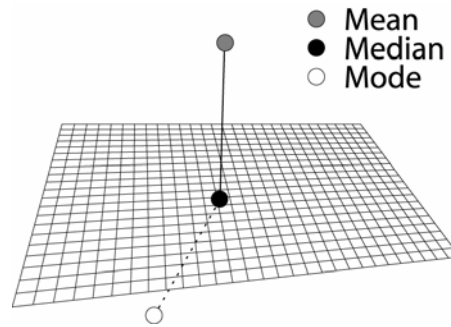
Total number of pixels scrutinised (ignoring edge pixels due to $5 \times 5$ windows)	15376
No. of pixels for which Mode $\rightarrow$ Mean longer than Median $\rightarrow$ Mean	13552
Percentage	88.14%
No. of pixels for which Mode classified on opposite of plane normal to Median $\square$ Mean	7932
Percentage	51.59%

**Table 6.2b. Pixel classification statistics for ‘Flowers’ image (original in Figure 6.4b).**

Total number of pixels scrutinised (ignoring edge pixels due to $5 \times 5$ windows)	15376
No. of pixels for which Mode $\rightarrow$ Mean longer than Median $\rightarrow$ Mean	13318
Percentage	86.62%
No. of pixels for which Mode classified on opposite of plane normal to Median $\rightarrow$ Mean	10661
Percentage	69.34%

**Table 6.2c. Pixel classification statistics for ‘Flowers’ image (original in Figure 6.4c).**

For between 54.5% and 69.3% of the total number of pixels, the angle between the median  $\rightarrow$  mode vector was more than  $90^\circ$  from the median  $\rightarrow$  mean vector, i.e. the modal value lies on the other side of the plane passing through the median and normal to the median  $\rightarrow$  mean vector (Figure 6.9). This confirms that in the majority of cases the ordering of the means is as expected with the mean and mode vectors being on opposite sides of the median.



**Figure 6.9. Relative positions of the three means and connecting vectors in 3-D.**

Table 6.3 shows similar results which were obtained for five  $128 \times 128$  control images (Figures 6.10a–e respectively). Of these images, two were basically homogeneous grey-level background images, with an intensity offset to prevent coincidence of the means within the intensity range 0–255 ( $R = G = B = 205$ ); two were basically homogeneous grey-level background images seated on the mean ( $R = G = B = 128$ ); and one was composed entirely of random input vectors – constituting essentially 100% impulse noise. The first four images were contaminated in various ways with Gaussian noise ( $\sigma^2 = 250, 1000$ ) and 70% impulse noise (see the caption to Figure 6.10). The two impulse noise examples – Figures 6.10d and 6.10e – illustrate respectively distributions that have an underlying mode in the midst of noise, and those that are truly random.

Gaussian noise experiments with larger variance at grey-level 205 were not possible because of errors introduced by truncation at the upper end of the intensity range, and impulse noise experiments were not conducted at grey-level 128 because of the greater probability of coincidence of the means, which would have hindered analysis.

The impulse noise results clearly show a decrease in the number of occurrences where the mode to mean distance exceeds that of the mode to median. This is expected for random distributions where the ordering of the means is more likely to be non-normal.

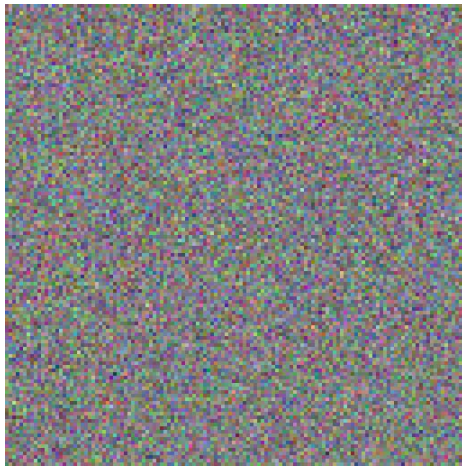




**Figure 6.10a.** Background intensity = 205, contaminated with Gaussian noise  $\sigma^2 = 250$ .



**Figure 6.10b.** Background intensity = 128, contaminated with Gaussian noise  $\sigma^2 = 250$ .



**Figure 6.10c.** Background intensity = 128, contaminated with Gaussian noise  $\sigma^2 = 1000$ .



**Figure 6.10d.** Background intensity = 205, contaminated with 70% impulse noise.



**Figure 6.10e.** Entire image composed of random colour vectors (essentially 100% impulse noise).

Total number of pixels scrutinised (ignoring edge pixels due $5 \times 5$ windows)	15376
No. of pixels for which Mode $\rightarrow$ Mean longer than Median $\rightarrow$ Mean	15234
Percentage	99.08%
No. of pixels for which Mode classified on opposite of plane normal to Median $\rightarrow$ Mean	8495
Percentage	55.25%

**Table 6.3a. Pixel classification statistics for noisy image Control 1 (Gaussian noise,  $\sigma^2=250$ , Intensity 205).**

Total number of pixels scrutinised (ignoring edge pixels due $5 \times 5$ windows)	15376
No. of pixels for which Mode $\rightarrow$ Mean longer than Median $\rightarrow$ Mean	15221
Percentage	98.99%
No. of pixels for which Mode classified on opposite of plane normal to Median $\rightarrow$ Mean	9021
Percentage	58.66%

**Table 6.3b. Pixel classification statistics for noisy image Control 2 (Gaussian noise,  $\sigma^2=250$ , Intensity 128).**

Total number of pixels scrutinised (ignoring edge pixels due $5 \times 5$ windows)	15376
No. of pixels for which Mode $\rightarrow$ Mean longer than Median $\rightarrow$ Mean	15241
Percentage	99.12%
No. of pixels for which Mode classified on opposite of plane normal to Median $\rightarrow$ Mean	8729
Percentage	56.77%

**Table 6.3c. Pixel classification statistics for noisy image Control 3 (Gaussian noise,  $\sigma^2=1000$ , Intensity 128).**

Total number of pixels scrutinised (ignoring edge pixels due $5 \times 5$ windows)	15376
No. of pixels for which Mode $\rightarrow$ Mean longer than Median $\rightarrow$ Mean	14686
Percentage	95.51%
No. of pixels for which Mode classified on opposite of plane normal to Median $\rightarrow$ Mean	12800
Percentage	83.24%

**Table 6.3d. Pixel classification statistics for noisy image Control 4 (Impulse noise, Intensity 205).**

Total number of pixels scrutinised (ignoring edge pixels due $5 \times 5$ windows)	15376
No. of pixels for which Mode $\rightarrow$ Mean longer than Median $\rightarrow$ Mean	12952
Percentage	84.24%
No. of pixels for which Mode classified on opposite of plane normal to Median $\rightarrow$ Mean	8705
Percentage	56.61%

**Table 6.3e. Pixel classification statistics for noisy image Control 5 (Impulse noise 100%).**

#### 6.4.4 NOISE REMOVAL

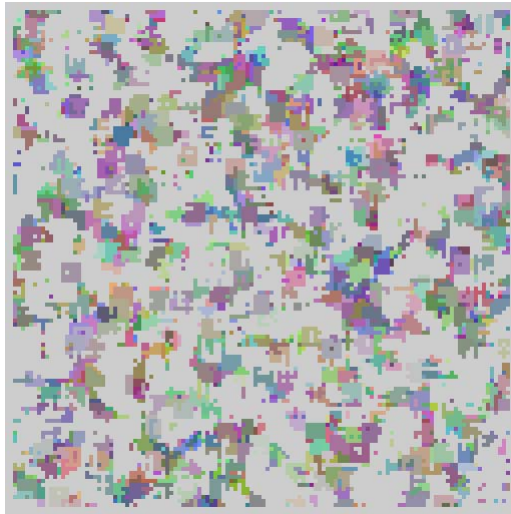
As stated earlier, it is well known that the mean provides an optimal estimate of a signal which is subject to a Gaussian noise distribution, while the median maximises the likelihood function for a double exponential noise distribution:<sup>74</sup> this gives the median better performance than the mean under impulse noise conditions. On the other hand, for noise distributions that are not Gaussian or double exponential, this simple analysis breaks down. However, it seems that the mode should be preferable in such cases, as it represents the most probable value of any distribution: specifically, when the form of the distribution is dictated more by the systematic variations in the underlying signal than by noise, the mode would be expected to be the best available estimator of the true signal level.



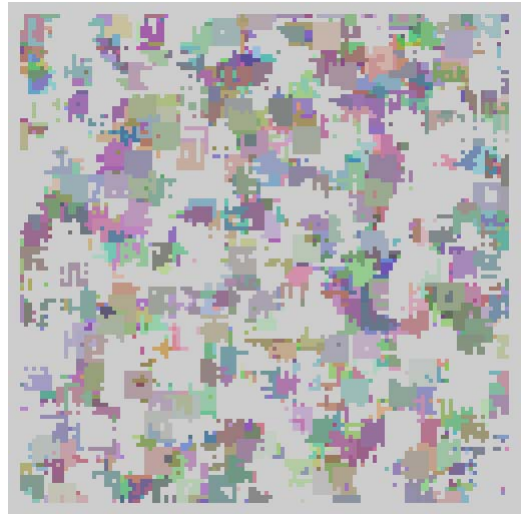
Perhaps the most widely used method of testing the efficacy of noise removal algorithms is to measure the normalised mean square error (NMSE) between the original and output images. However, it would be misleading and uninformative to compare the NMSE results for the median- and mode-filtered natural images already shown: as the mode filter takes the majority vote of a distribution it has the effect of removing fine detail and smoothing image irregularities. This cumulative effect of many small errors might appear to imply that the new filter will not be effective at removing noise spikes and should be regarded more as an image enhancement tool. However, the action of the mode filter is such that it can eliminate any number of outliers at either end of a distribution, so it should actually be more effective at removing noise than the median filter, which can remove a maximum of 50% of points at either end of a distribution.

We have found that for low to moderate noise, the median filter performs quite adequately and similarly to the mode filter, and deviates less from the original image. However, when the filter window contains two or more noisy pixels that are not positioned symmetrically in the distribution, the position of the median will move and thus the output will be sub-optimal, as already shown in Figure 6.4a. To show this effect and to provide quantitative results avoiding the problems mentioned above, Figure 6.10d is utilised once more. A constant-level image is used to discount the detail-smoothing effects of both types of filter and allow examination solely of the noise removal properties. The results of applying the median and mode filters to this image are shown in Table 6.4, and depicted in Figures 6.11a–d and 6.12a–d respectively. Both filters show a decrease in noise level as the filter window size increases, but the advantage of a large mode filter window in the presence of extremely high noise levels (i.e. those that exceed 50% of the filter window input) becomes absolutely obvious (see especially the last line in Table 6.4).

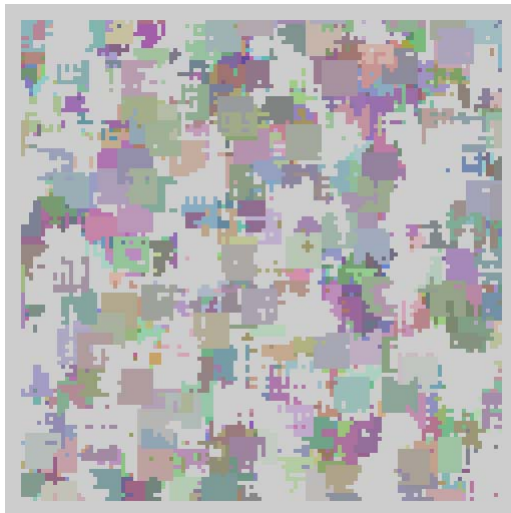
It is now clear that although the median filter returns better noise figures as the window size increases, there is a distinct trade-off between the per-pixel magnitude difference and the size of the resultant noise patch, whereas the mode filter replaces the noisy pixel in a more thoroughgoing fashion.



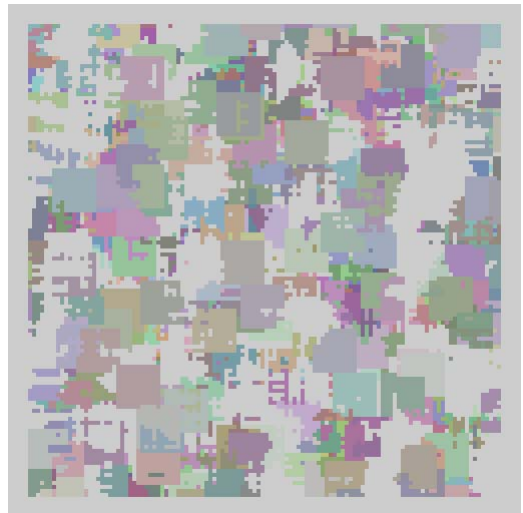
**Figure 6.11a.** Figure 6.10d after  $5 \times 5$  VMF.



**Figure 6.11b.** Figure 6.10d after  $7 \times 7$  VMF.



**Figure 6.11c.** Figure 6.10d after  $9 \times 9$  VMF.



**Figure 6.11d.** Figure 6.10d after  $11 \times 11$  VMF.

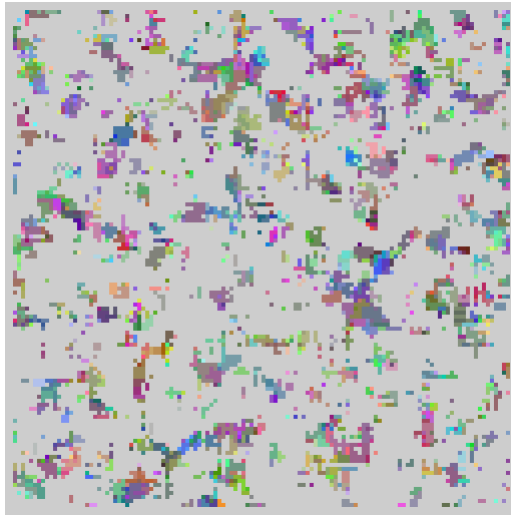


Figure 6.12a. Figure 6.10d after  $5 \times 5$  TVMF.

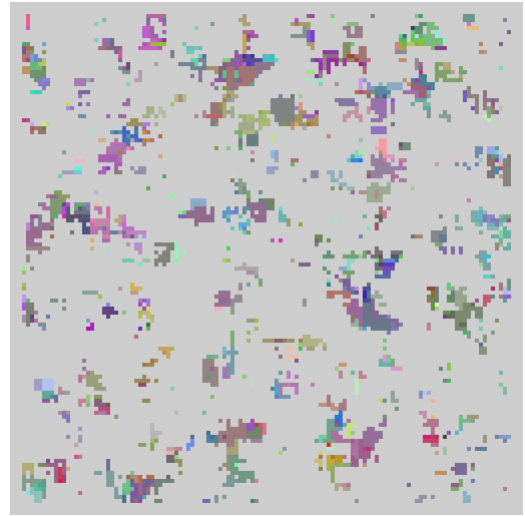


Figure 6.12b. Figure 6.10d after  $7 \times 7$  TVMF.

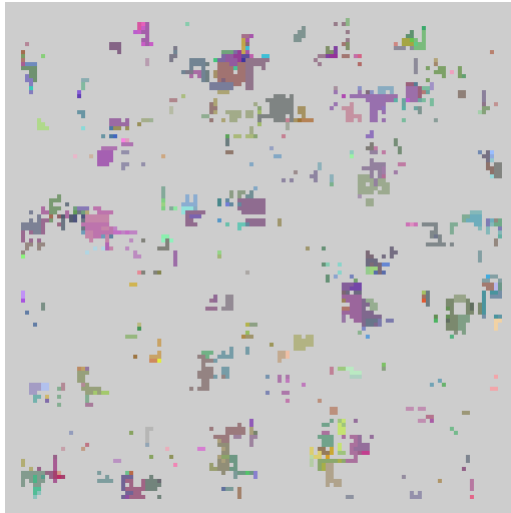


Figure 6.12c. Figure 6.10d after  $9 \times 9$  TVMF.

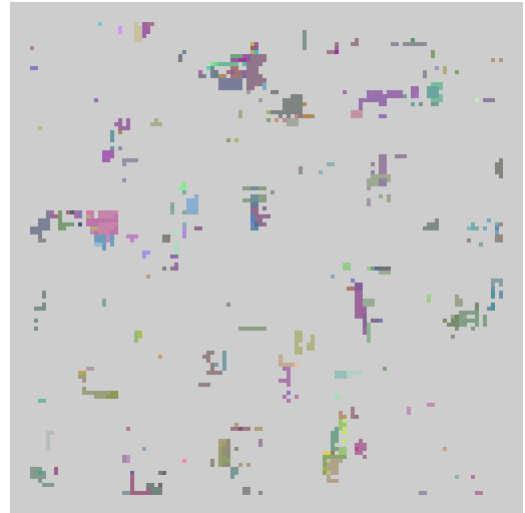


Figure 6.12d. Figure 6.10d after  $11 \times 11$  TVMF.

Window size (square neighbourhood)	Normalised Mean Square Error	
	Median	Mode
$5 \times 5$	$4.9385 \times 10^{-2}$	$4.1776 \times 10^{-2}$
$7 \times 7$	$4.1601 \times 10^{-2}$	$2.3517 \times 10^{-2}$
$9 \times 9$	$3.8960 \times 10^{-2}$	$1.3059 \times 10^{-2}$
$11 \times 11$	$3.7758 \times 10^{-2}$	$6.962 \times 10^{-3}$

Table 6.4. Normalized errors for various window sizes for Figure 6.10d.

For qualitative comparison, Figures 6.4a–c were also contaminated with 70% impulse noise and processed using  $9 \times 9$  median and mode filters, as shown in Figures 6.13a–i.

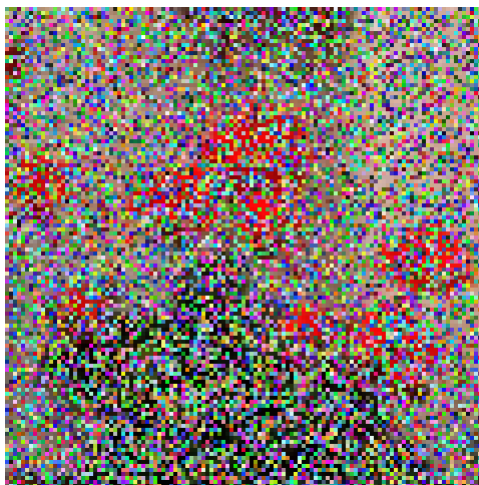
The claim is not that the new mode filter restores noisy images more closely to the original than the median; rather, it is that in cases of high contamination, salient image features such as object boundaries, base colours or intensity can still be picked out with greater success with the new algorithm than with the basic vector median filter. The above examples amply demonstrate this.

## 6.5 EDGE SHIFTING CHARACTERISTICS OF MODE FILTERS

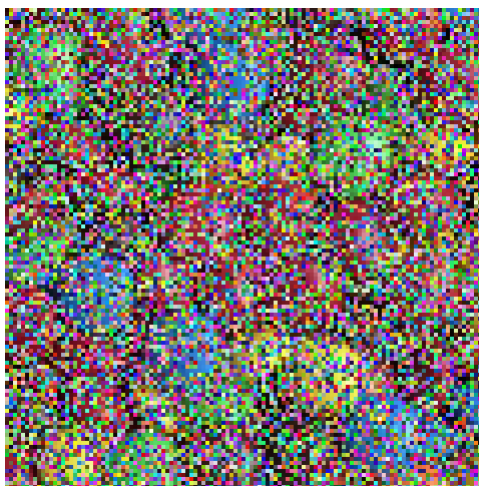
As mentioned in Section 4.4, Section 6.3 above and noted by Astola et al.,<sup>74</sup> a multidimensional filter should produce the same results as the corresponding scalar filter when applied to a 1-D signal. It can be seen that for a line of any orientation in a representative 3-D colourspace along which lie the data points of interest (a 1-D signal), the described vector method of truncating according to radius and location is analogous to rank-ordering and truncating according to position in one dimension, as given in the method of Davies.<sup>134</sup> To confirm this experimentally, we use two approaches.

First, the ‘Peppers’ test image was converted to 1-D greyscale (by averaging the  $R$ ,  $G$ ,  $B$  values), then subsequently converted back to the RGB colourspace (with  $R = G = B$ ), and the results of the scalar mode compared with the vector mode. Testing on a pixel-by-pixel basis revealed an exact correspondence between the pixel intensities, and the two images can be seen to produce identical characteristics when compared with the original (Figure 6.14a–c).

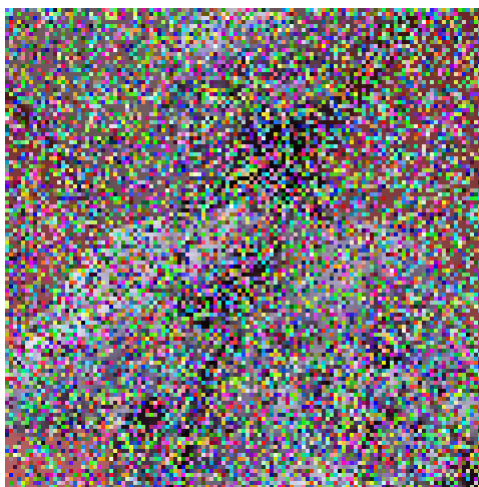
Second, previous experimental work by Davies was extended to determine the greyscale edge shift introduced by both the scalar and vector truncated median filters.<sup>69,119,136</sup> It is shown that the edge shift introduced by the vector filter is the same as that of the scalar, demonstrating the exact correspondence between the two operators in 1-D.



**Figure 6.13a. Figure 6.4a contaminated with 70% impulse noise.**



**Figure 6.13b. Figure 6.4b contaminated with 70% impulse noise.**



**Figure 6.13c. Figure 6.4c contaminated with 70% impulse noise.**

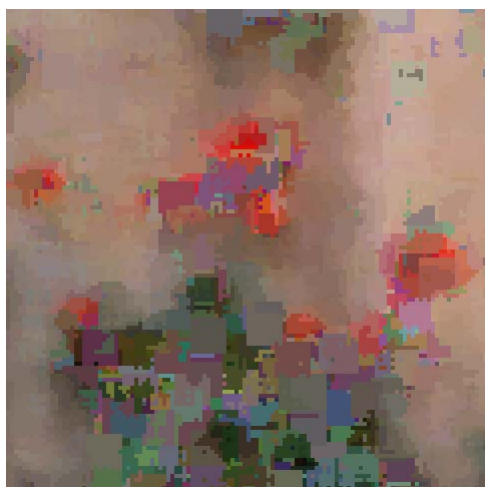


Figure 6.13d. Figure 6.13a after  $9 \times 9$  VMF.

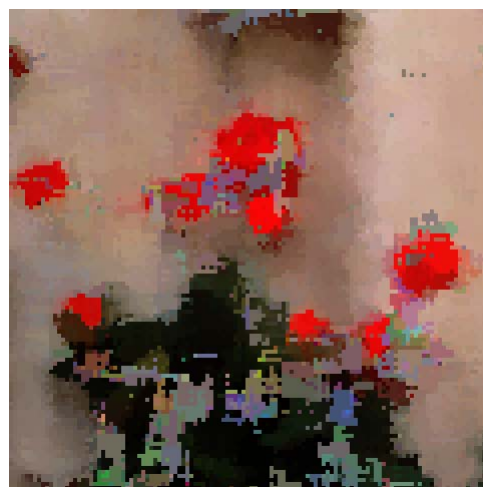


Figure 6.13g. Figure 6.13a after  $9 \times 9$  TVMF.

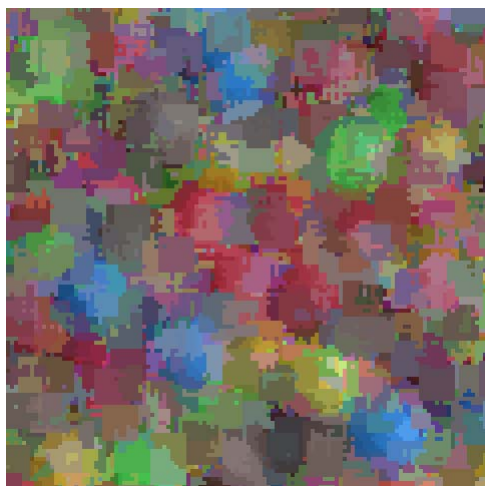


Figure 6.13e. Figure 6.13b after  $9 \times 9$  VMF.

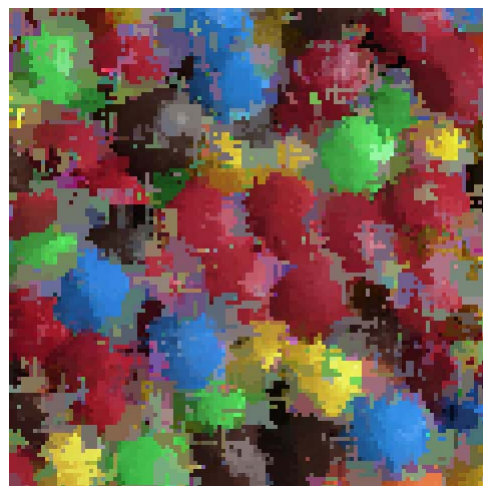


Figure 6.13h. Figure 6.13b after  $9 \times 9$  TVMF.

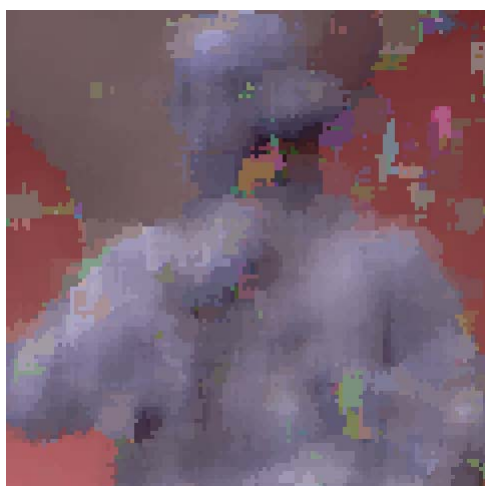


Figure 6.13f. Figure 6.13c after  $9 \times 9$  VMF.



Figure 6.13i. Figure 6.13c after  $9 \times 9$  TVMF.



**Figure 6.14a.** ‘Peppers’ image converted from RGB to greyscale.



**Figure 6.14b.** Figure 6.14a after  $5 \times 5$  TMF.

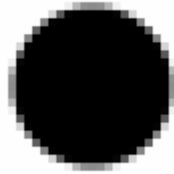


**Figure 6.14c.** Figure 6.14a after  $5 \times 5$  TVMF.

To achieve this, a number of greyscale circles of varying curvature were taken and both types of truncated median filter were applied in nearly circular windows and the differences in area before and after filtering were measured. The circles are generated at sub-pixel accuracy, each pixel being divided into a further  $5 \times 5$  array, and the proportion of those within the circle is used to give a mean greyscale value for the whole pixel. To further improve accuracy additional measurements were taken at all possible subpixel positions of the circle centres on a  $0.2 \times 0.2$  pixel lattice so that 25 results are obtained for each generated circle, and average the results.



A typical generated circle (inverted here for ease of scrutiny) takes the form:



Note that the method of measuring the sizes of the processed circles is to measure their integrated intensities, and thus deduce the new effective radius values.

Figure 6.15a–e shows the application of the scalar and vector truncated median filters to circles of curvature 0.1 to 0.9 for  $5 \times 5$  to  $13 \times 13$  circular filters respectively. For comparison, each graph includes the edge shift results for scalar and vector median filters. Notice that the asymptotic shift (equal to  $1/C_0$ ) corresponds exactly to the value of the circle radius.

As already discussed in Chapter 4, the vector median filter produces identical results to those for the scalar algorithm, and as predicted, application of the vector ‘mode’ filter on a 1-D data set produces the same results as for the scalar truncated median filter.

The results obtained for the various edge curvatures and filter window sizes show three main regions of interest:

1. a region where the mode operator closely tracks the median,
2. a region where the mode departs markedly from the median, and
3. a region of asymptotic decline from the maximum edge shift.

These results will be dealt with in order as Cases 1–3 overleaf; a minor perturbation will also be pointed out under the heading Case 4. First, we note the results of previous theoretical work which predicted the edge shifts that will be introduced by mean, median and mode filters.<sup>69,119,136,137</sup> these results are summarised in Table 6.5.



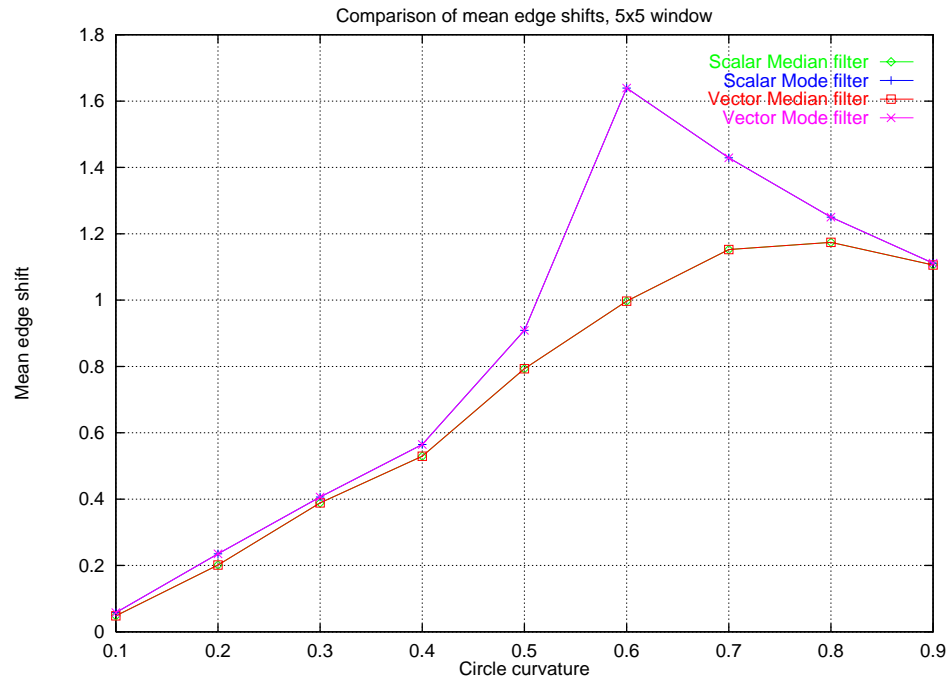


Figure 6.15a. Application of the scalar and vector truncated median filters to circles of curvature 0.1 to 0.9 in 0.1 steps for  $5 \times 5$  circular filter, against results obtained from scalar and vector median filters (note identical results for scalar and vector versions of each filter).

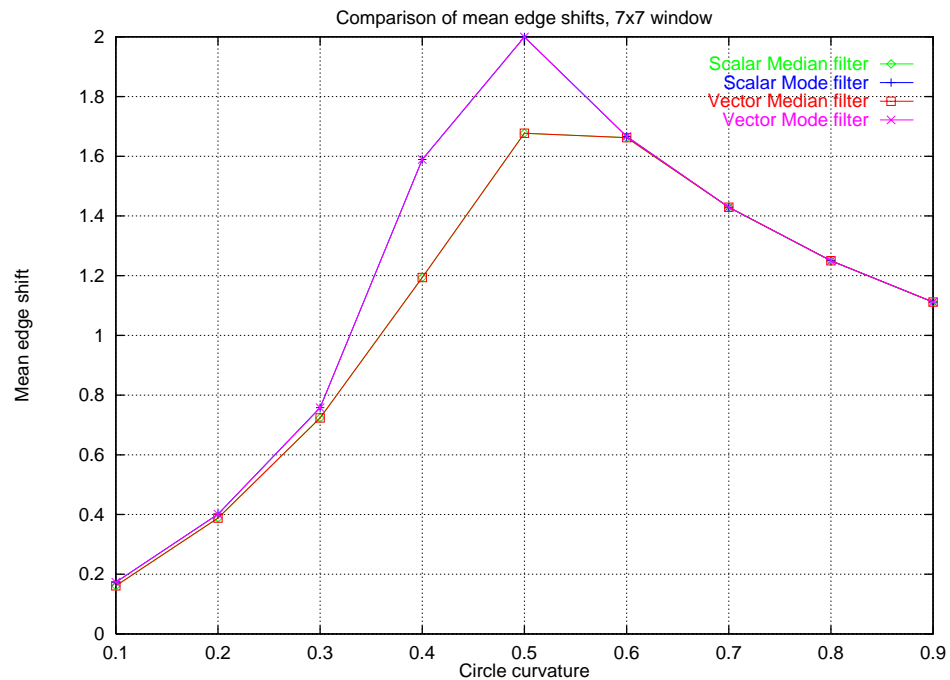


Figure 6.15b. Application of the scalar and vector truncated median filters to circles of curvature 0.1 to 0.9 in 0.1 steps for  $7 \times 7$  circular filter, against results obtained from scalar and vector median filters.

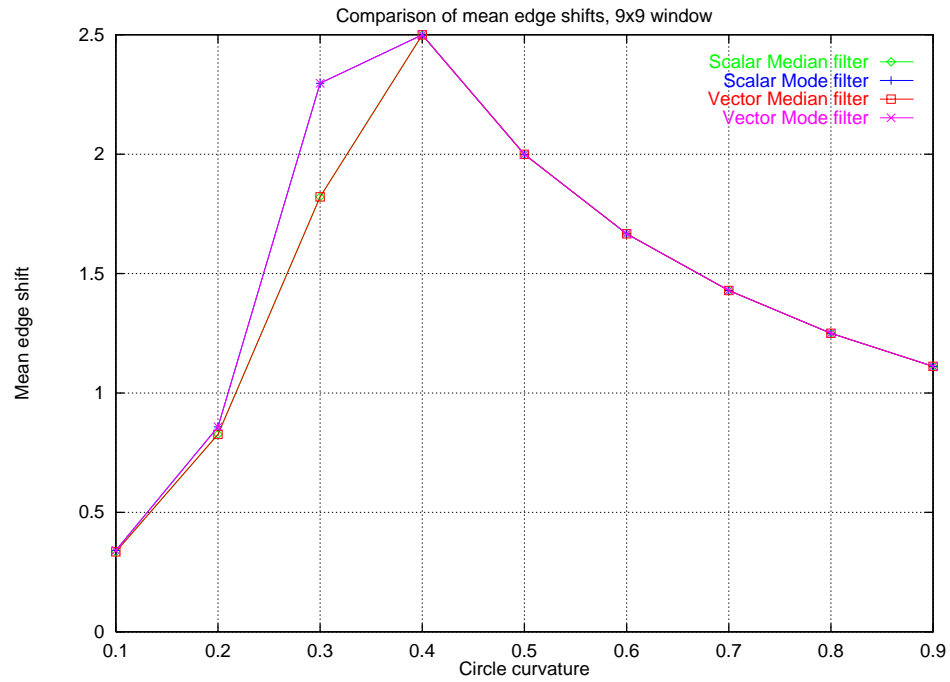


Figure 6.15c. Application of the scalar and vector truncated median filters to circles of curvature 0.1 to 0.9 in 0.1 steps for  $9 \times 9$  circular filter, against results obtained from scalar and vector median filters.

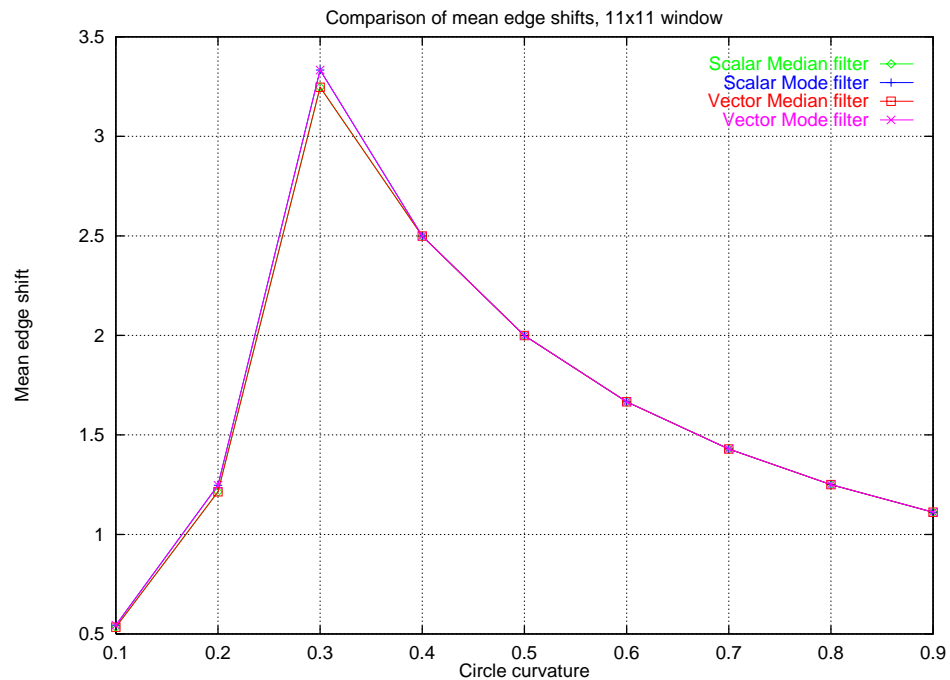
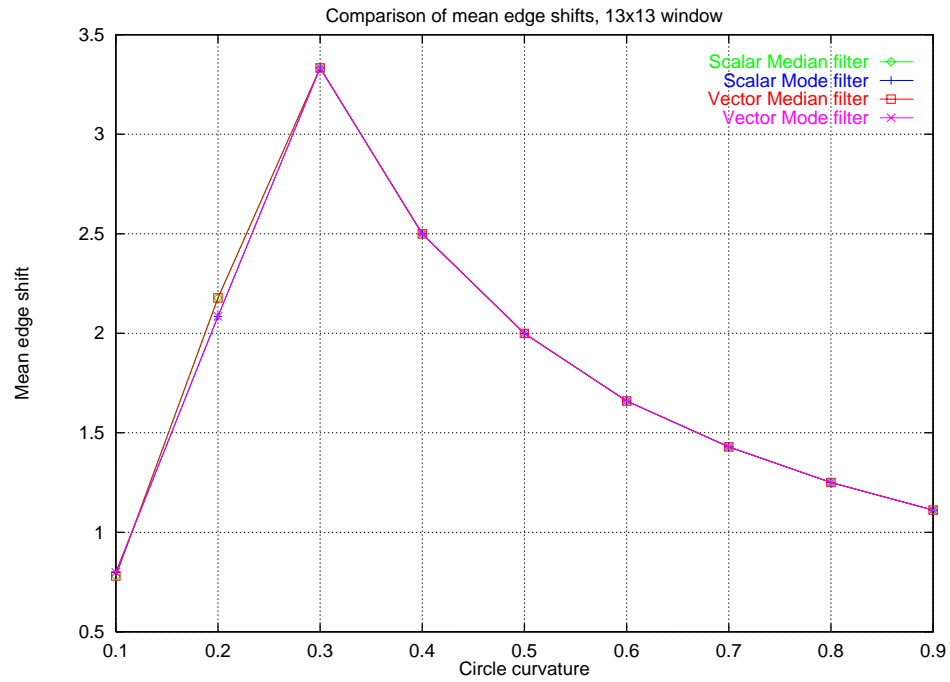


Figure 6.15d. Application of the scalar and vector truncated median filters to circles of curvature 0.1 to 0.9 in 0.1 steps for  $11 \times 11$  circular filter, against results obtained from scalar and vector median filters.



**Figure 6.15c.** Application of the scalar and vector truncated median filters to circles of curvature 0.1 to 0.9 in 0.1 steps for  $13 \times 13$  circular filter, against results obtained from scalar and vector median filters.

Edge type	Filter		
	Mean	Median	Mode
Step	$1/6C_c a^2$	$1/6C_c a^2$	$1/6C_c a^2$
Intermediate	$\approx 1/7C_c a^2$	$1/6C_c a^2$	$1/2C_c a^2$
Linear slant	$1/8C_c a^2$	$1/6C_c a^2$	$1/2C_c a^2$

**Table 6.5.** Summary of previous edge shift calculations. In this table,  $C_c$  is the curvature of the edge being examined, and  $a$  is the radius of the operator window.

### 6.5.1 CASE 1

When the object curvature is low, i.e. when the filter window is small compared with the circle, the numbers of background and foreground pixels are large compared with those on the edge. In this case the contents of the operator window can be taken to approximate a step edge and results similar to those for the median are obtained.<sup>136</sup> As the mode operator tends to maintain the background or foreground intensity right up to the edge, more intermediate-valued boundary pixels are removed than may be the case with the median filter, so a slightly greater shift occurs. (This factor will be considered in more detail in Section 6.6.) It is clear from Figure 6.15 that both the median and mode filter follow the  $\frac{1}{6} C_c a^2$  rule quite well up to the point of departure for the mode (see Case 2 below).

To proceed further, it was found that greater accuracy was required, and the experiments were repeated in the region of interest for the five operator window sizes, as shown in Figure 6.16.

### 6.5.2 CASE 2

The rapid increase in mean edge shift (departing from that of the median) represents the transition from a step edge to a slant edge; thus the edge shift increases from  $\frac{1}{6} C_c a^2$  to a maximum of  $\frac{1}{2} C_c a^2$  (Table 6.5). The curvature at which this occurs depends very much on the relative curvatures  $C_c$ ,  $C_w (= 1/a)$  of the circle and operator window. However, it is clear that rapid change will occur when the curvature of the object under scrutiny is similar to, or greater than, that of the operator window.

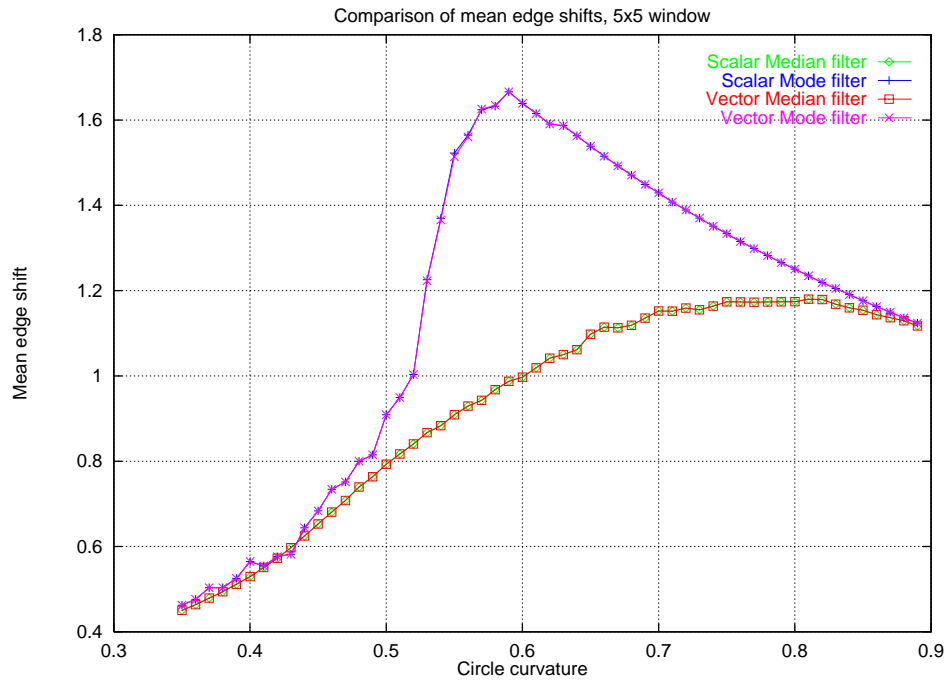


Figure 6.16a. Application of the scalar and vector truncated median filters to circles of curvature 0.1 to 0.9 in 0.01 steps for  $5 \times 5$  circular filter, against results obtained from scalar and vector median filters.

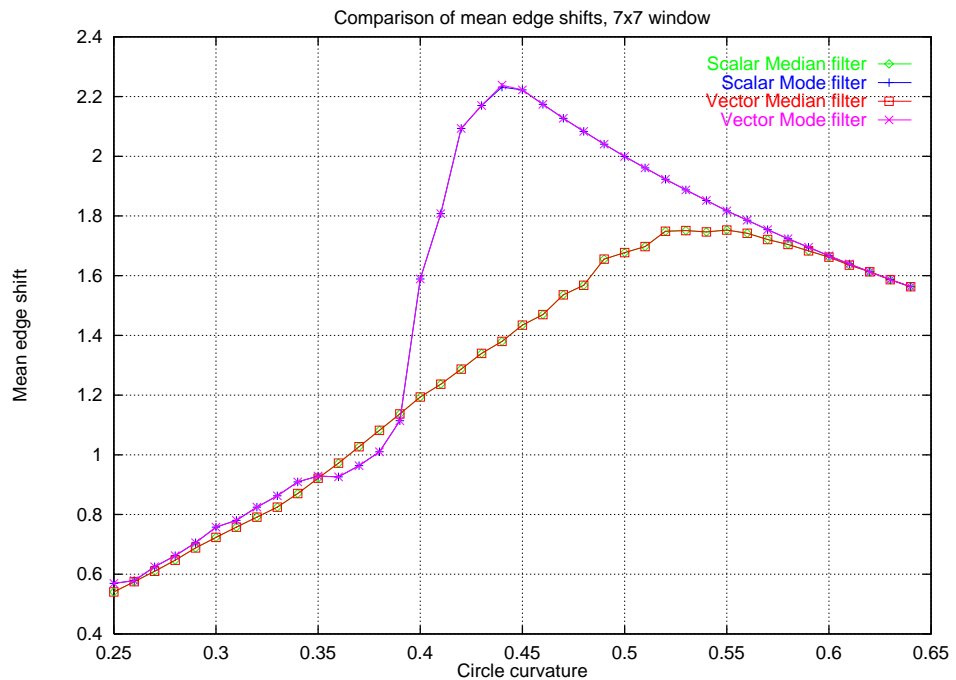


Figure 6.16b. Application of the scalar and vector truncated median filters to circles of curvature 0.1 to 0.9 in 0.01 steps for  $7 \times 7$  circular filter, against results obtained from scalar and vector median filters.

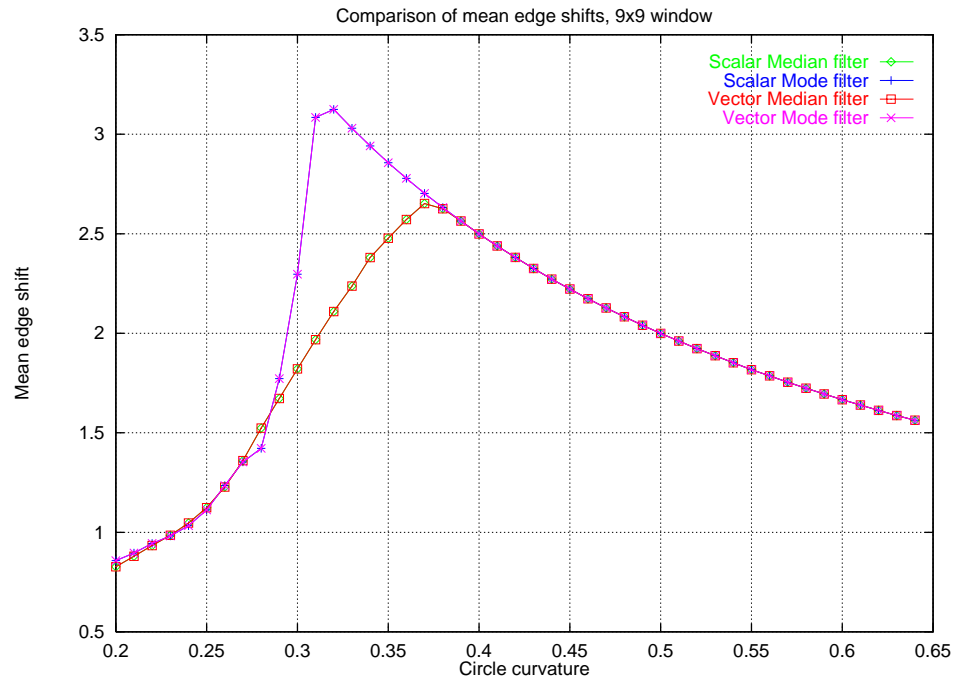


Figure 6.16c. Application of the scalar and vector truncated median filters to circles of curvature 0.1 to 0.9 in 0.01 steps for  $9 \times 9$  circular filter, against results obtained from scalar and vector median filters.

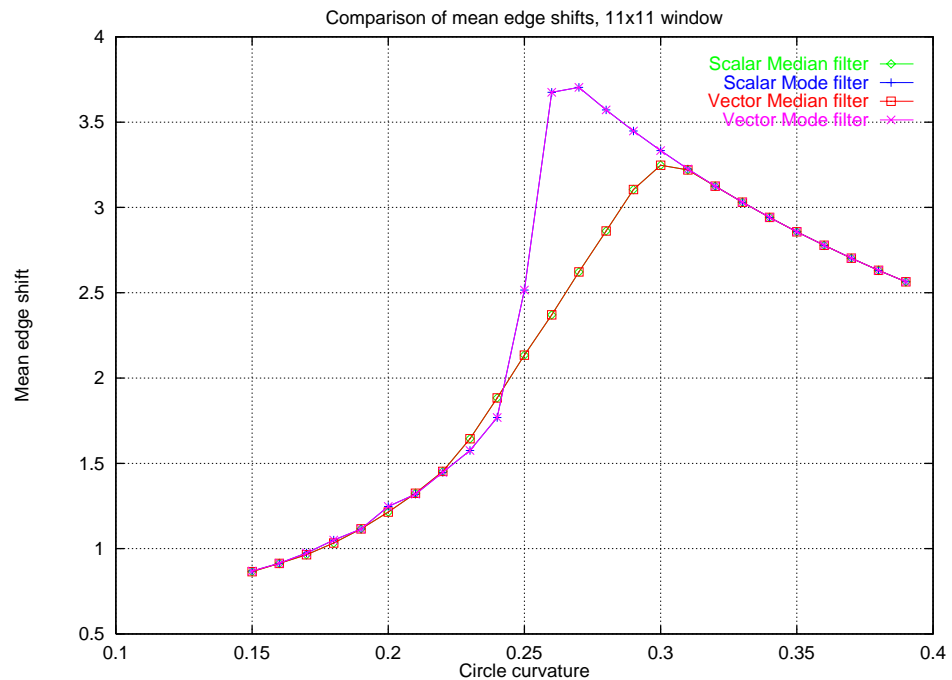


Figure 6.16d. Application of the scalar and vector truncated median filters to circles of curvature 0.1 to 0.9 in 0.01 steps for  $11 \times 11$  circular filter, against results obtained from scalar and vector median filters.

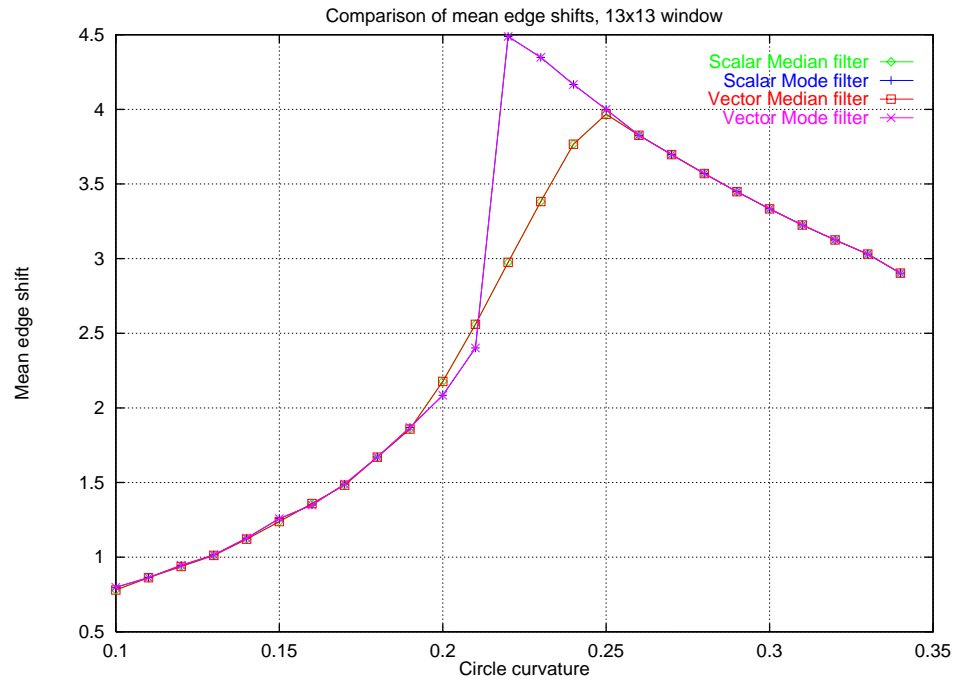


Figure 6.16e. Application of the scalar and vector truncated median filters to circles of curvature 0.1 to 0.9 in 0.01 steps for  $13 \times 13$  circular filter, against results obtained from scalar and vector median filters.

Window size (circular neighbourhood)	Step- to Intermediate-edge transition		
	$C_w$	$C_c$	Ratio $C_c/C_w$
$5 \times 5$	0.400	0.50	1.250
$7 \times 7$	0.286	0.40	1.400
$9 \times 9$	0.222	0.30	1.350
$11 \times 11$	0.182	0.25	1.375
$13 \times 13$	0.154	0.212	1.378

Table 6.6. Characterization of rapid onset of mode changes.

Table 6.6 shows the values of  $C_c$  for which the mode filter begins its rapid increase in edge shift, the critical threshold being taken as 10% difference relative to the median value\*. To proceed, the ratio  $C_c/C_w$  is calculated: taking the mean of the results, the transition is characterised when the object curvature  $C_c$  is approximately 35% greater than that of the window. Discrepancies arise because of the discrete nature of the data; for instance a circular  $5 \times 5$  operator derived by truncating border pixels has an area of 21 pixels as opposed to the continuous operator which has an area of 19.2 pixels. Thus an increase in the accuracy of the sub-pixel measurement method, and weighting operator window pixels to more closely reflect an ideal continuous operator, would be expected to yield slightly more consistent results. However, an exact theory would have to relate to the digitised nature of the lattice – e.g. see Davies (1999).<sup>119</sup>

### 6.5.3 CASE 3

As the edge shift acts towards the origin of curvature, the action of the mode operator is characterised by the steady removal of boundary pixels until a point is reached where the entire object is eliminated. This occurs when the ratio of background pixels to foreground pixels *excluding boundary pixels of intermediate grey-level* (which the mode filter will largely ignore) is greater than unity. At this point, any object of up to half the area of the operator window will be wholly removed. This is borne out by the close match to the empirical results of Figure 6.16, as shown in Table 6.7. Circles of higher curvature (smaller area) are of course also removed, as indicated by the asymptotic maximum (for which the shift equals the circle radius) in Figure 6.16. Note that these effects are essentially identical for the mode and median: this is made clear by the close agreement that occurs at the high curvature regions of Figures 6.16c–e. However, this model is rather simplistic in that it does not consider the intensity gradient at the

---

\* While 10% is arbitrary, it gives a useful indication of when a significant change has occurred: smaller variations would be subject to noise; larger variations would give greater shift and would erode the quantity that we are aiming to measure.



Window size (circular neighbourhood)	Point at which total object removal occurs (area in pixels)	
	Expected	Measured (curvature)
$5 \times 5$	9.82	9.02 (0.59)
$7 \times 7$	19.24	16.23 (0.44)
$9 \times 9$	31.81	30.68 (0.32)
$11 \times 11$	47.52	43.09 (0.27)
$13 \times 13$	66.37	64.91 (0.22)

**Table 6.7. Object removal measurements.**

circumference of the generated test circles. A more rigorous treatment is given below in Section 6.6.

#### 6.5.4 CASE 4

It is also interesting to note the behaviour around the point where the curvature of the operator window is the same as that of the circle: for filters greater than  $5 \times 5$ , the mode filter exhibits very slightly lower mean edge shift than for the median. This appears to be due to a number of factors: first, recall the ‘edge-crispening’ effect noted by Davies,<sup>17</sup> which has the effect of tending to maintain the dominant object intensity closer to the boundary than for the median filter; second, the truncated median filter always errs on the ‘safe’ side, truncating if anything fewer of the intensity values than would be the case for an idealised continuum intensity distribution. Indeed, as described in Section 6.3, when the filter is applied to an even symmetric distribution it selects the vector closest to the old median: on the circumference of a circle, where the edge shift acts towards the origin of curvature, the old median must necessarily have been a point inside the circle, thus acting further to minimise the apparent edge shift. Examination of the generated test circles show very many cases where such even symmetric distributions arise.

Finally, the method by which the edge shift is calculated also has some bearing on the results obtained: Figure 6.17 shows an example of two of the generated circles and the results of applying median and mode filters to each of them; it can be seen that although the radius of the median-filtered circle is greater than that of the mode-filtered circle, the pixels at the circumference are of various intermediate grey-levels. When integrating the circle pixel intensity it thus appears that the mode-filtered circle exhibits a lower mean edge shift because the circle pixel intensity is maintained right to the edge, thus leading to a higher aggregate intensity.

In summary, this measured effect occurs because of a combination of the designed characteristics of the filter, the symmetry and the discrete nature of the artificially generated test data, and the method by which the edge shift is measured; in other words it is a pathological effect obtained due to the nature of the experiment and is unlikely to be observable in normal machine vision applications.

## **6.6 FINITE EDGE WIDTH MODEL: CALCULATION OF BREAKPOINTS**

It will be clear from the discussions in the previous section that a rather complicated situation exists and that a thoroughgoing discrete model would be required before the performance of the mode filter could be evaluated exactly. Nevertheless, Section 6.5 brought out a number of relevant points, some of which related to a continuum approximation. In this section we try to make these points more concrete by observing that the variations shown in Figures 6.15 and 6.16 can be regarded as being comprised of significant regions separated by 'breakpoints'. For the present purpose we identify three breakpoints: the lowest is the point (bp1) where the mode curve separates from the median curve and takes a sharp turn upwards; the second is the point (bp2) where the mode curve stops rising and starts falling steadily; the third is the point (bp3) where the *median* curve stops rising and starts falling steadily.

We have found that we can explain these breakpoints qualitatively, and to a satisfactory extent quantitatively, by modelling the intensity profiles of the circular

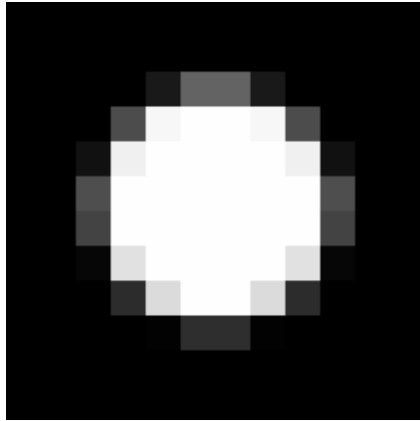


Figure 6.17a.

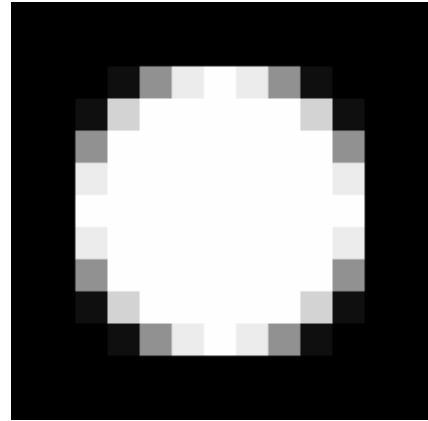


Figure 6.17d.

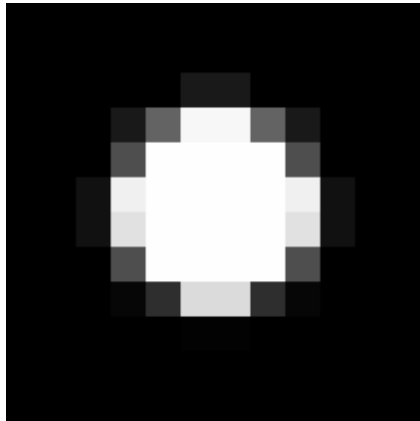


Figure 6.17b.

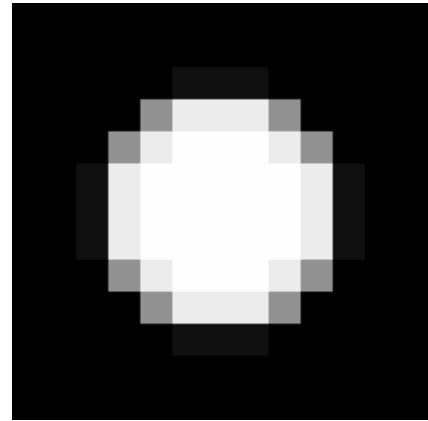


Figure 6.17e.

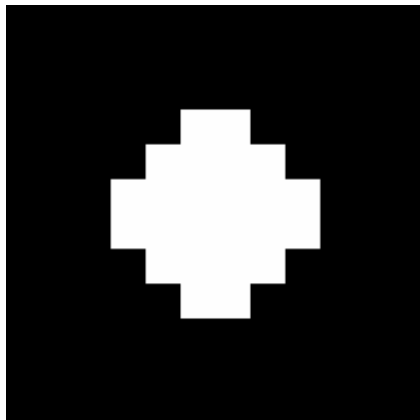


Figure 6.17c.

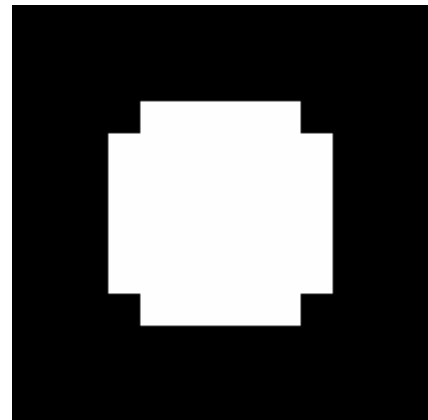


Figure 6.17f.

**Figure 6.17.** Two examples of generated test circles to measure mean edge shift. (a) Generated circle of curvature 0.3, with a horizontal offset of 4 sub-pixels and a vertical offset of 5 sub-pixels. (b) Application of  $5 \times 5$  circular median filter. (c) Application of  $5 \times 5$  circular truncated median filter. (d) Generated circle of curvature 0.22, with zero horizontal and vertical offset. (e) Application of  $9 \times 9$  circular median filter. (f) Application of  $9 \times 9$  circular truncated median filter.

objects as having linear slopes bounded by constant background and foreground regions (Figure 6.18a). Because of the discrete nature of the lattice and the limited number of available grey levels, and the necessarily narrow width of the edge region (which *a priori* must be around 1–2 pixels), such a model can only be an approximation. Nevertheless, it immediately leads to some useful conclusions when examined in conjunction with the experimental results.

The first such conclusion is that, when applying a median filter in practice, the circles will not disappear until a significantly higher curvature has been reached than would *a priori* have been imagined. This is because the smaller radius  $b$  actually leads to a higher outer radius value  $b_{med}$  (Figure 6.18a) because of the linear intensity profile, and it is this value that must be inserted as the limiting value into the formula for the limiting size of circle before the circle finally disappears:

$$\pi b_{med}^2 = \pi a^2 - \pi b_{med}^2 \quad (6.3)$$

$$\therefore b_{med} = \frac{a}{\sqrt{2}} \quad (6.4)$$

Taking the overall width of the intensity gradient at the edge as  $t$ , we obtain the following relations for the inner and outer edge radii,  $b_{mod}$  and  $b_{med}$  respectively:

$$b_{mod} = b - \frac{t}{2} \quad (6.5)$$

$$b_{med} = b + \frac{t}{2} \quad (6.6)$$

We can now use the experimentally observed median breakpoint value  $bp3$  to estimate  $t$ :

$$\frac{t}{2} = \frac{a}{\sqrt{2}} - b \quad (6.7)$$

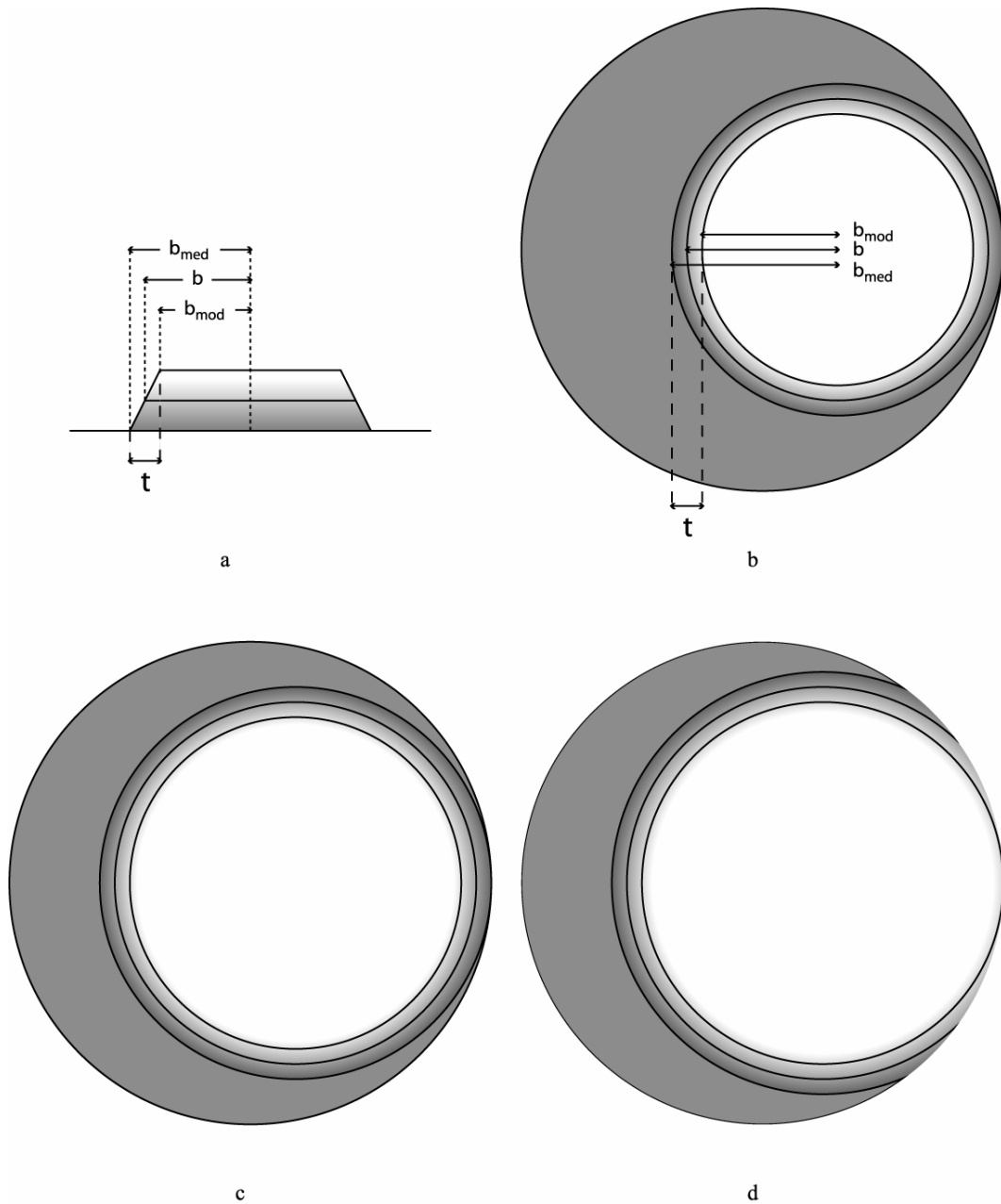


Figure 6.18. Situation pertaining at the three breakpoints. Each diagram shows the window (the circle of largest radius  $a$ ) and a circular object of basic radius  $b$ , with inner ( $b_{mod}$ ) and outer ( $b_{med}$ ) radii reflecting the width  $t$  of the linear sloped edge region. The shading represents the local intensity (e.g darkest = 0, lightest = 255). In (a) an overhead view is shown. In (b), the object is about to disappear under the action of a median filter, the area within the outer radius ( $b_{med}$ ) being exactly half the area of the window. In (c) the object is about to disappear under the action of a mode filter, the area within the inner radius ( $b_{mod}$ ) being exactly equal to that outside the outer radius ( $b_{med}$ ). In (d) the innermost and outermost regions are again the same area. These diagrams lead to estimation of breakpoints bp3, bp2, bp1 respectively (see text).

If the model is a useful one, the values of  $t$  obtained from the five curves will be consistent: otherwise the opposite will be true. Taking the effective radii calculated from the pixel areas of the different sized operators and inserting them into Equation 6.7, we obtain a narrow range of values (Table 6.8) with a mean of 1.45 and a remarkably low standard deviation of 0.041. With this degree of reliability we can go on to use this edge model to estimate the breakpoints for the mode filter.

The upper mode breakpoint (bp2) is relatively easy to calculate, using the following formula:

$$\pi b_{mod}^2 = \pi a^2 - \pi b_{med}^2 \quad (6.8)$$

which reflects the fact that the region in the window outside the outer edge must equal the region within the inner edge, at the point when the mode filter is about to eliminate the whole circular object (Figure 6.18b).

Substituting for  $b_{med}$  and  $b_{mod}$  leads to the formula:

$$a^2 = b_{med}^2 + b_{mod}^2 = 2b^2 + \frac{t^2}{2} \quad (6.9)$$

from which we can deduce bp2 as given by:

$$b = \sqrt{\frac{a^2}{2} - \frac{t^2}{4}} \quad (6.10)$$

Finally, bp1 is estimated as follows: it is the point at which the whole of the inner edge of the circular object lies just within the window, while at the same time the area within the inner edge must equal the area outside the outer edge (Figure 6.18c). Unfortunately, the latter area cannot be written down in the form of a simple formula, and has to be estimated numerically. Table 6.9 gives the values of bp1 obtained in this way, and also shows the values obtained from the mode curves in Figure 6.16. It also shows the corresponding results for bp2.

Window size	Number of active pixels <sup>*</sup>	Effective radius $a^{\dagger}$	bp3 curvature (radius $b$ )	$t$
$5 \times 5$	21	2.585	0.895 (1.117)	1.422
$7 \times 7$	37	3.432	0.595 (1.681)	1.492
$9 \times 9$	69	4.687	0.385 (2.597)	1.433
$11 \times 11$	97	5.557	0.310 (3.226)	1.407
$13 \times 13$	137	6.604	0.255 (3.922)	1.496

<sup>\*</sup> The corners of the windows were truncated to make them approximate more closely to circles.

<sup>†</sup> The effective radii were calculated from the active areas of the windows.

**Table 6.8. Calculation of finite edge width  $t$ .**

window	bp1		bp2		bp2 – bp1	
	observed	estimated	observed	estimated	observed	estimated
$5 \times 5$	0.50	0.49	0.59	0.60	0.09	0.11
$7 \times 7$	0.40	0.38	0.45	0.43	0.05	0.05
$9 \times 9$	0.30	0.29	0.32	0.31	0.02	0.02
$11 \times 11$	0.25	0.24	0.27	0.26	0.02	0.02
$13 \times 13$	0.212	0.207	0.220	0.217	0.008	0.010

**Table 6.9. Breakpoints for the mode filter.**

In view of the crudeness of the model (limitation to linear slope, continuum approximation to a discrete lattice, and so on<sup>\*</sup> – as indicated above), the degree of agreement between estimated and observed values is good, while that for the differences between the two breakpoint values (last two columns in Table 6.9) is even better – presumably because any errors in the estimation of  $t$  and in the meaningfulness of the model as a whole tend to cancel in the latter case. Finally, note that, for the mode filter at least, this represents a zero parameter fit – i.e. the

<sup>\*</sup> Recall also the simple 10% criterion used in obtaining the observed values of bp1, as discussed earlier in relation to Table 6.

only assumptions made are those relating to linearity of the edge profile (including disregard of discrete pixel effects): the single parameter that is adjusted relates to fitting the median rather than the mode.

These observations confirm that the behaviour of the mode filter is understood both qualitatively and, to a good degree, quantitatively, and make it clear that significantly more accurate assessment of the situation must await detailed discrete calculations – as in the case of the median filter.

## **6.7 CONCLUDING REMARKS**

This section has examined the properties of the truncated median filter cited in earlier work as providing a useful implementation of the mode filter.<sup>134</sup> It has confirmed the value of this type of filter for providing an enhancement capability when applied to grey-scale images, and it has also examined the possibility of extending the concept to the processing of colour images. A procedure has been developed – the ‘truncated vector median filter’ (TVMF) – based on vector median filtering, which leads to a highly effective image enhancement capability for colour images. The procedure, which like the vector median filter is able to resist colour bleeding problems, reduces exactly to the truncated median filter for grey-scale images, thereby lending support to its validity. Further support comes from the excellent noise-suppression capability of the extended filter, which is able to cope with well over 50% outlier noise – thereby considerably exceeding the 50% limit of the widely used median and vector median types of filter.<sup>17</sup>

The edge shifting properties of the TVMF filter have also been explored and compared with those of the vector median filter. For low boundary curvatures it appears that the TVMF filter follows the vector median filter very closely, whereas above a certain curvature level the TVMF filter introduces a substantially higher level of edge shift, though this is in line with the edge shift theory of Davies.<sup>119,136</sup> At high curvature levels the mode shifts again merge with those for the vector median filter – ultimately because the small objects leading to these high curvatures



are progressively eliminated when they lie largely inside the filter window. It is important to note that these properties of the mode filter are due to the fundamental nature of a mode filter, namely that it should take most account of the majority intensity within the window, even if this means shrinking objects around their periphery: this is expected behaviour rather than the result of *ad hoc* programming. Indeed, far from the mode filter corresponding to *ad hoc* programming, it represents a highly important type of algorithm that is considerably more robust (in an outlier suppression sense) than more widely used types of filter.

Clearly, if filters with particularly low edge shifts are required, mode filters should not be used as their specification is quite different; in such circumstances it is in any case known that median filters should be replaced by specially designed variants, such as hybrid median<sup>135</sup> or neural based filters<sup>110</sup> that have been specially tailored to the task.

# Chapter 7: Application of Neural Networks to Colour Image Filtering

## 7.1 APPLICATION TO GREYSCALE IMAGES

The subject of artificial neural networks was introduced in Chapter 4 and an outline given of the wide range of machine vision applications to which they have been applied successfully. Particular emphasis was given to the multilayer perceptron with error backpropagation (MLP), the architecture employed by Davies and Greenhill<sup>110</sup> with great efficacy on noisy greyscale images. Noise filtering of images using neural networks is also discussed in Greenhill,<sup>142</sup> Nightingale and Hutchinson,<sup>143</sup> Weber et al,<sup>144</sup> Klimasauskas<sup>145</sup> and Pham and Bayro-Corrochano.<sup>146</sup>

Here the work of Greenhill is extended to investigate how well the MLP performs in the domain of colour images contaminated using the same noise models used in Chapter 4.

It should be noted that although Greenhill demonstrated the usefulness of neural networks at removing noise, the results presented suffered from some significant drawbacks. Firstly, the training set consisted of only two images, namely overlapping runner beans on a homogeneous background; the majority of presentations to the neural network will have been either bean texture or background texture and thus these images represent a rather specialized image class. Secondly, although three noise models were used, the use of ‘salt and pepper’ noise (i.e. where a noisy pixel  $n=0$  or  $n=255$  for an 8-bit greyscale image) is not particularly useful; this type of noise can be easily rejected without removing a great deal of image detail by using an extremum-type filter for noise levels of less than 20% or so. Of the two remaining noise models, the impulse noise was only represented by 7.6%

contamination (random value between 0 and 255), and the Gaussian model only used a variance of 100. Thirdly, although the testing set included a more general natural image, the ANN's performance on this image was marginally better than a  $3 \times 3$  median filter, as distinct from the obviously superior performance on the images of runner beans. This is no doubt due to the discrepancy between the classes of training and testing image, the network having 'learnt' how to represent bean textures free of noise, but not that of the tree in the natural image. Finally, only two local window sizes were the subject of the experiments ( $3 \times 3$  and  $5 \times 5$ ), possibly restricting the performance of the ANN.

The aim of this chapter is to establish that ANNs, specifically MLPs, are effective at removing noise from colour images and that their employment may be considered a viable alternative to the time-consuming task of traditional filter design. This is demonstrated by considering a number of factors discussed below.

## **7.2 ARCHITECTURE OF MLP**

The values of the pixels in the local window are first normalized by dividing by 255, then each value in each colour channel is fed to a separate neuron in the input layer; thus experiments with  $3 \times 3$  square windows would have 27 input neurons,  $5 \times 5$  square windows would have 75 neurons, etc. Pixel pre-processing is kept to a minimum to observe how the neural network performs when presented with the raw image data. In all experiments there are three output neurons, representing the replacement image vector at the centre of the sub-image window.

Greenhill utilized a network with two hidden layers; whilst this may speed convergence and aid the solving of certain problems,<sup>17</sup> a single hidden layer is capable of achieving any desired modelling error criterion.<sup>147</sup> For this reason only one hidden layer is used here, although the number of neurons is varied in the experiments, corresponding to sub-image windows of different sizes.

Window Size	Pixels in Window	Input Neurons	Hidden Neurons	Output Neurons
$3 \times 3$	9	27	6	3
$5 \times 5$	21	63	8	3
$7 \times 7$	37	111	11	3
$9 \times 9$	69	207	15	3
$11 \times 11$	97	291	17	3

**Table 7.1. Number of neurons in each layer for respective sub-image window sizes.**

Setting the number of hidden neurons is often quite arbitrary; too few neurons and it is unlikely that the network will produce acceptable results, too many and the network will have difficulty generalizing beyond the training set.<sup>148</sup> Setting the number of hidden neurons to approximately the square root of the number of input neurons is often used as a starting point.<sup>148,149</sup>

In a paper by Ciftcioglu and Türkcan which presents a model for deriving the number of hidden neurons using statistical tests, the anecdotal example they give produced best results with between  $\sqrt{n}$  and  $0.5n$  for  $n$  input neurons.<sup>150</sup>

The action of the ANN should ideally be isotropic; to this end circular windows were employed by excluding corner pixels. Each sub-image window should be presented to the ANN in all possible orientations but in the first instance this would significantly increase the computational burden. Two of the training sets were selected for their variety of edge types and orientations to ameliorate this omission.

Initial experiments showed setting the learning rate  $\eta$  to 0.2, momentum parameter  $\alpha$  to 0.1 (see Eq. 4.45, Section 4.8.2) and the number of hidden neurons to  $\sqrt{n}$  for  $n$  input neurons to be reasonable compromise between convergence rate and minimum error. Table 7.1 shows the network sizes used in the experiments. In all cases the weights were initialized to random values between -0.3 and 0.3 before training commenced.

The hypothesis is that the performance of the ANN will increase with respect to the size of the input window; should this not be the case the use of larger windows will not decrease the ANN's effectiveness as non-contributory pixels will have their weights set to zero, though this may increase training time.

### **7.3 TRAINING SET**

The ANN must be presented with a representative training set of the images on which it is ultimately to be employed; if the network is trained on natural images with a predominance of certain spatial and chromatic characteristics its final trained weights will reflect this. Its noise rejection performance on artificial or industrially derived images may be sub-optimal or indeed detrimental – this is perhaps most clearly illustrated by an example in which the network was trained to remove noise from the 'Lena' training image (Figures 7.1a), and the trained network subsequently applied to remove noise from the 'Peppers' test image (Figure 7.1b). The result (Figure 7.1c) shows that trained network weights reflect the predominance of low saturation colours and skin tones; the similarity in colour characteristics between the images in Figures 7.1a and 7.1c is obvious.

Clearly, chromatic distortion such as this is undesirable, more so than the removal of noise. To combat this, either the network must be:

- trained on a disparate set of images so that it may generalize,

or

- be restricted to use on particular image class to work with a select set of spatial and chromatic qualities.

The latter may in fact be beneficial for industrial inspection, for instance.

With this in mind, three different training and testing sets were determined to investigate these observations; the aim is to demonstrate that training sets which belong to a definite image class will elicit better results from the ANN than a disparate image set. For all image sets, images were contrast-stretched prior to use.



**Figure 7.1.** Effect of training neural network on a specialized image: (a) shows the training image, (b) shows the test image and (c) the resultant output.

### 7.3.1 STANDARD IMAGES

The well-known Waterloo Repertoire Colorset image database<sup>151</sup> was chosen as a class of dissimilar training images for the ANN. Unlike the other two training sets, there are only eight images in the Waterloo Repertoire Colorset; thus the testing set also contains eight different images to test the generalization ability of the ANN. It is expected that the ANN will perform poorly at this task, so the difference in the

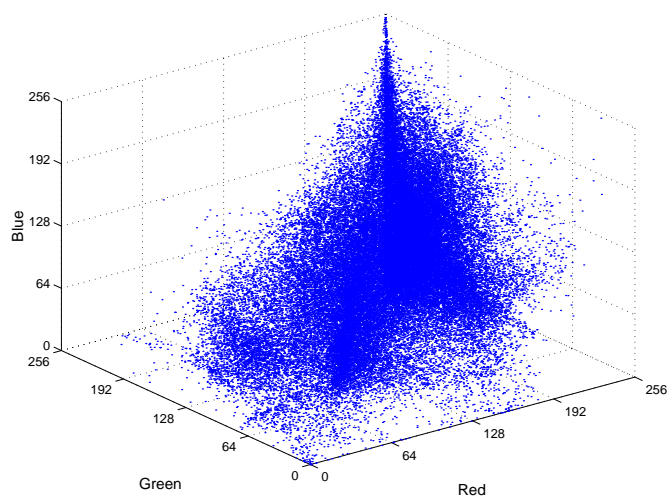
number of training and testing images compared to the other experimental datasets will have little effect on demonstrating this. The dissimilar nature of the images is given in Figure 7.2a-c showing RGB scattergrams of the first three images in the set, ‘Clegg’, ‘Frymire’ and ‘Lena’.

### **7.3.2 PLASTIC COMPONENT IMAGES**

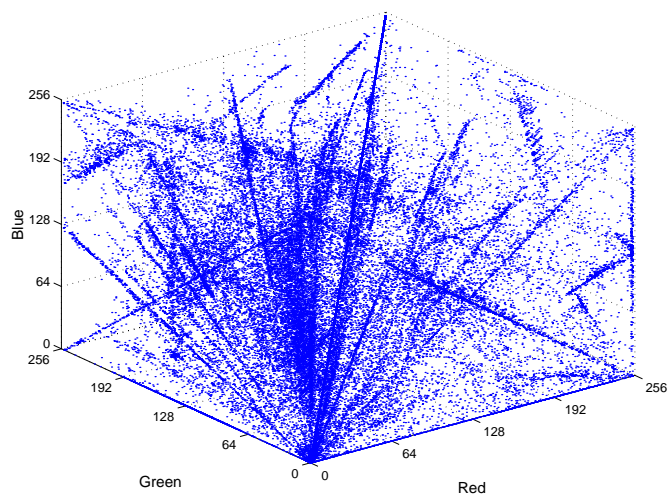
The plastic toy image set was chosen for its similarity to mechanical parts involved in industrial inspection containing diverse features, namely:

- Reflections
- Shadows
- Holes
- Corners
- Different orientations
- Limited range of colours
- Transparent portions

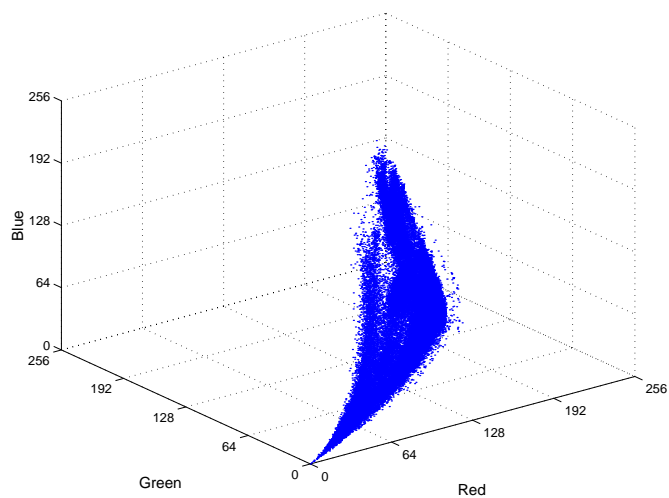
Sixteen images were used for training and sixteen for testing. RGB scattergrams of three of the training images are shown in Figure 7.3a-c.



**Figure 7.2a.** RGB scattergram of 'Clegg' image.



**Figure 7.2b.** RGB scattergram of 'Frymire' image.



**Figure 7.2c.** RGB scattergram of 'Lena' image.



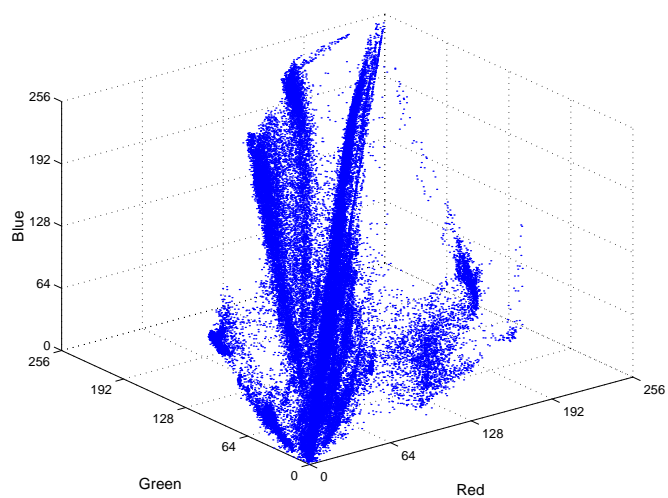


Figure 7.3a. RGB scattergram of plastic component training image 6.

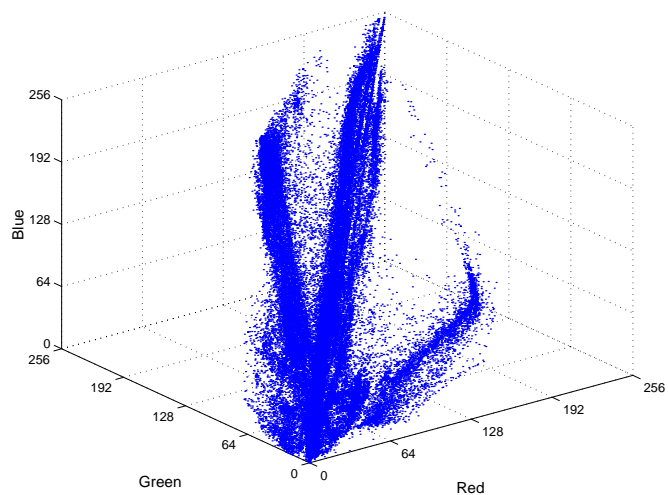


Figure 7.3b. RGB scattergram of plastic component training image 7.

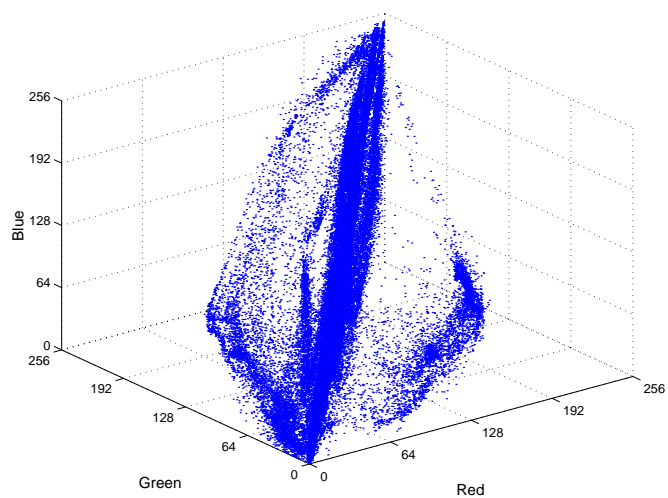


Figure 7.3c. RGB scattergram of plastic component training image 8.

### **7.3.3 CONFECTIONERY IMAGES**

This image set has comparable qualities to those described above, but the images are somewhat more self-similar. The pieces of confectionery are all the same size, the edges are predominantly circular and there is less chromatic variance. This is clearly illustrated by the scattergrams in Figure 7.4a-c.

## **7.4 EXPERIMENTAL RESULTS**

It can be seen from Figures 7.2 – 7.4 that all the images (with the exception of the artificially generated images, ‘Clegg’, ‘Frymire’ and ‘Serrano’) are strongly correlated with intensity which is to be expected (see Chapter 2). In view of this, experiments were also conducted in the Ohta and LAB colourspaces to investigate whether training the ANN in a decorrelated or a more perceptually accurate colourspace affected the training or performance of the network. As with experiments carried out in Chapter 4 and Chapter 5, the networks were trained on two impulse, two Gaussian and two mixed-noise types of contamination. As experiments regarding the efficacy of ANNs for noise removal have not been previously carried out for colour images, it is not known how long training may be expected to take to achieve acceptable results; a secondary aim of this work is to determine how much training is necessary. Therefore, it was decided for each network that training was concluded after 50 complete passes of each image set, corresponding to approximately 26 million presentations for the standard image set and 52 million for the two others.

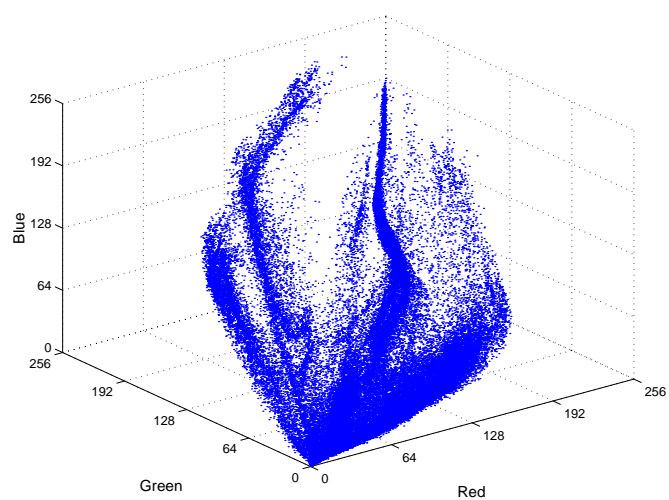


Figure 7.4a. RGB scattergram of confectionery training image 1.

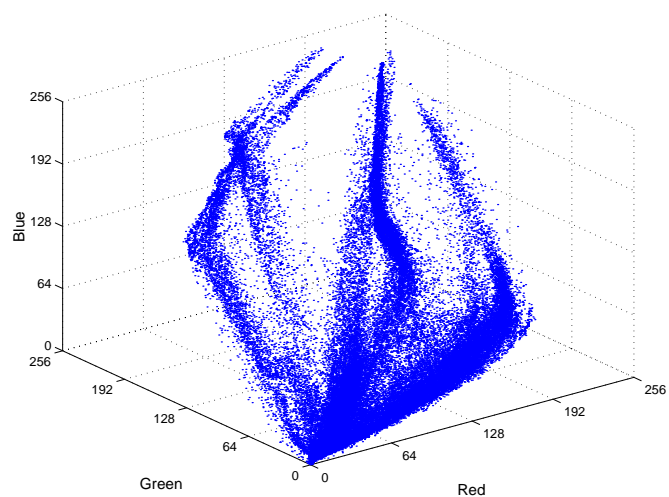


Figure 7.4b. RGB scattergram of confectionery training image 2.

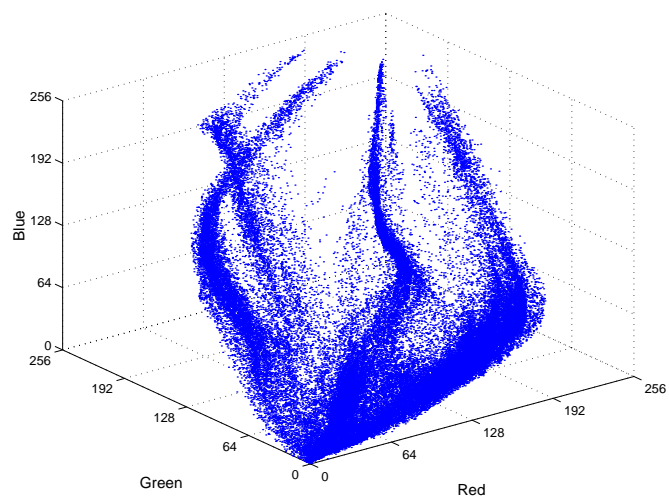


Figure 7.4c. RGB scattergram of confectionery training image 3

### **7.4.1 TRAINING**

For each noise type, the total squared error over all the training images for each pass was measured. The results show broadly similar traits; in the interests of brevity, only a few example sets of results are shown here. The complete set of training results may be found in Appendix C1.

#### **STANDARD IMAGES**

For the standard image noise experiments, the choice of colourspace has the most significant effect on the training error, followed by the window size used. The RGB colourspace is the worst performer in all cases with the exception of the 2% impulse noise experiment where the RGB  $9 \times 9$  and  $11 \times 11$  operators return an error between the Ohta  $3 \times 3$  and  $5 \times 5$  operators at their respective minimums. The difference between the Ohta and LAB operators is not so great, with the LAB  $11 \times 11$  operator giving the lowest final training error for the impulse noise experiments and the Ohta  $9 \times 9$  and  $11 \times 11$  giving best scores for all the other training noise types.

The training error becomes fairly static after the first 20 passes for the poorer performing networks; they have either become trapped in local error minima (unlikely across all experiments) or they have converged on the best solution for their respective parameters. For the best performing networks, the error is still decreasing; further training may increase their efficacy. The training error for the 5% impulse experiment is shown in Figure 7.5.

#### **PLASTIC COMPONENT IMAGES**

As with the standard image set, the choice of colourspace has the most significant effect on the training error, followed by the window size used. For the impulse noise experiments the RGB colourspace returns the greatest error, and the LAB and Ohta colourspace errors are close for all window sizes, barring the Ohta  $3 \times 3$

network. No RGB network was able to match the low training errors achieved by the other colourspaces. The network with the lowest training error for 2% impulse contamination was the Ohta  $11 \times 11$ , and the Ohta  $9 \times 9$  for the 5% impulse noise case.

For the Gaussian noise experiments there is greater variation between the window size and colourspace used, but again the RGB colourspace is outperformed by both Ohta and LAB for all window sizes in both experiments. The largest Ohta windows give the lowest training error overall.

For the mixed noise cases, the Ohta colourspace gives the lowest error for the light mixed noise experiment with negligible difference between the  $7 \times 7$ ,  $9 \times 9$ , and  $11 \times 11$ . For the moderate mixed noise experiment, the Ohta and LAB networks perform comparably for a given operator size, the  $11 \times 11$  performing the best.

Similar observations to the standard image set may be made regarding the rate of error decrease; static after approximately 20 passes for the majority of networks but still decreasing for the better performers. Figure 7.6 shows the training errors for the heavy Gaussian noise experiment.

## **CONFECTIONERY IMAGES**

The training errors for the confectionery images show similar trends to those already discussed above.

For the impulse noise experiments, the difference between the Ohta and LAB colourspaces is negligible, but the lowest error is given by the LAB  $9 \times 9$  network; the Ohta colourspace gives the best results for the Gaussian and mixed noise cases ( $11 \times 11$  operator); Figure 7.7 shows the training error for the heavy mixed noise experiment. The RGB colourspace once more displays the greatest training error.

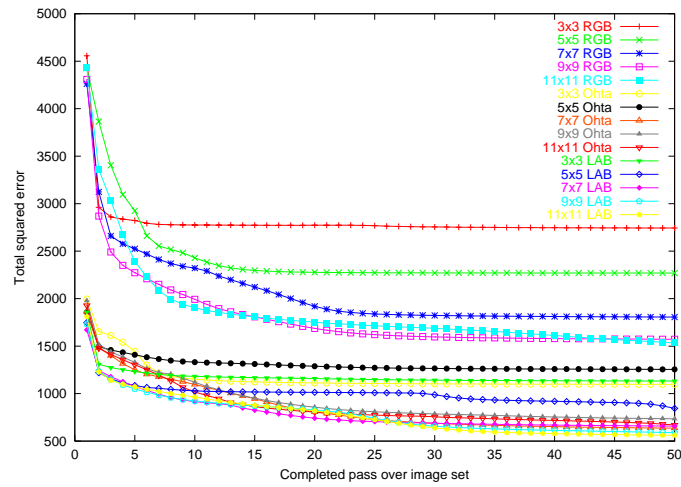


Figure 7.5. Training error for 5% impulse noise, standard image set.

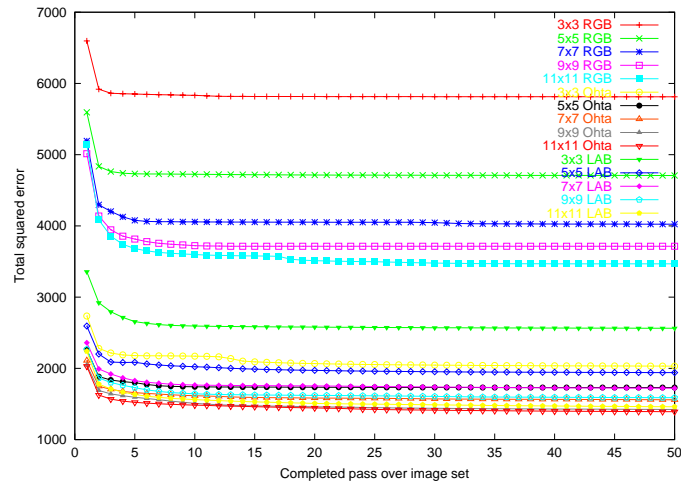


Figure 7.6. Training error for  $\sigma^2=1000$  Gaussian noise, plastic component image set.

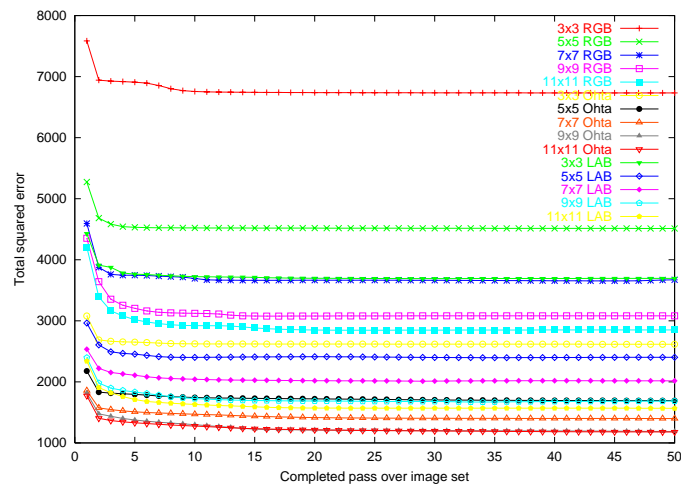


Figure 7.7. Training error for 5% and  $\sigma^2=1000$  mixed noise, confectionery image set.

### **7.4.2 TESTING**

Testing was conducted by converting each noisy image to the appropriate colour space, applying the neural network, then converting back to RGB and obtaining a NMSE measure. A second set of images for each image set was used to confirm each network's ability to generalize within an image class.

During training, the state of each network was saved every five passes over each image set; this ensures that during testing the 'best' set of weights can be identified.

For each noise type, the optimum performance is determined by measuring the NMSE summed over all the training and testing images for each saved network. For brevity, the same selected example for the training results is shown for each image set; the full results are given in Appendix C2.

Tables 7.2 – 7.4 list the best performing networks by noise type for each image set. It is interesting to note that in general the best performance is obtained from the final testing pass for the majority of cases; this indicates that further training may well lead to increased noise rejection performance. The conspicuous exceptions to this are the heavy Gaussian noise experiments, in which the best performance is obtained after only 5 or 10 passes. It was noted in Chapter 4 that the optimal operator in a Gaussian environment is the simple arithmetic mean filter – for the ANN to converge so quickly it is evident that the network is more closely operating as a complex linear filter rather than a non-linear classifier (in which training is necessarily longer to obtain the best classification boundary).

#### **STANDARD IMAGES**

It is immediately apparent when considering Table 7.2 that the testing results differ markedly from what one is led to expect from the discussion in Section 7.4.1. Rather than the RGB ANNs being the poor performers, the LAB ANNs return the higher error scores; this is true for all noise models.

As Table 7.2 shows, it is the Ohta colourspace using the smaller windows which produces the best results, the notable exception being the 2% impulse noise model in which the RGB ANNs perform the best. Figure 7.8 shows the testing results of the 5% impulse noise experiment.

### **PLASTIC COMPONENT IMAGES**

The testing results for the plastic component images are somewhat more mixed with Ohta and RGB networks performing similarly. The LAB networks again performed poorly compared to the larger operators in the other colourspace. With the exception of the heavy mixed noise model, the RGB networks gave the best error scores (Table 7.3). Figure 7.9 shows the testing results of the heavy Gaussian noise experiment.

### **CONFECTIONERY IMAGES**

The observations made regarding the plastic component images apply to the confectionery testing results also: the poorest performing are again the LAB and the smaller  $3 \times 3$  and  $5 \times 5$  window RGB ANNs, and the best performing are the larger window Ohta and RGB networks. The similarity in results between Table 7.3 and Table 7.4 is clear, with  $9 \times 9$  and  $11 \times 11$  window RGB ANNs dominating. Figure 7.10 shows the testing results of the heavy mixed noise experiment.

### **7.4.3 COMMENT ON COLOURSPACE SUITABILITY**

Some comment should be made regarding the differences between the error scores of the three colourspace obtaining during training and those obtained during the testing. Necessarily, the training and testing error results are measured in different ways; recall Eq. 4.46 which measures the training error as the sum-squared difference between the network output and the target output. For each colourspace this is done natively, i.e. no conversion is done back to RGB space at this stage.



Noise Model	ANN Colourspace	Window Size	Best Training Pass
2% impulse	RGB	$7 \times 7$	50
5% impulse	Ohta	$7 \times 7$	50
$\sigma^2=100$ Gaussian	Ohta	$3 \times 3$	50
$\sigma^2=1000$ Gaussian	Ohta	$5 \times 5$	10
2% and $\sigma^2=100$ mixed	Ohta	$7 \times 7$	50
5% and $\sigma^2=1000$ mixed	Ohta	$7 \times 7$	50

**Table 7.2. Best performing neural networks on standard image set.**

Noise Model	ANN Colourspace	Window Size	Best Training Pass
2% impulse	RGB	$9 \times 9$	50
5% impulse	RGB	$11 \times 11$	30
$\sigma^2=100$ Gaussian	RGB	$9 \times 9$	45
$\sigma^2=1000$ Gaussian	RGB	$9 \times 9$	5
2% and $\sigma^2=100$ mixed	RGB	$9 \times 9$	50
5% and $\sigma^2=1000$ mixed	Ohta	$9 \times 9$	30

**Table 7.3. Best performing neural networks on plastic component image set.**

Noise Model	ANN Colourspace	Window Size	Best Training Pass
2% impulse	RGB	$9 \times 9$	20
5% impulse	RGB	$11 \times 11$	45
$\sigma^2=100$ Gaussian	RGB	$11 \times 11$	50
$\sigma^2=1000$ Gaussian	Ohta	$11 \times 11$	5
2% and $\sigma^2=100$ mixed	RGB	$9 \times 9$	50
5% and $\sigma^2=1000$ mixed	Ohta	$9 \times 9$	30

**Table 7.4. Best performing neural networks on confectionery image set.**

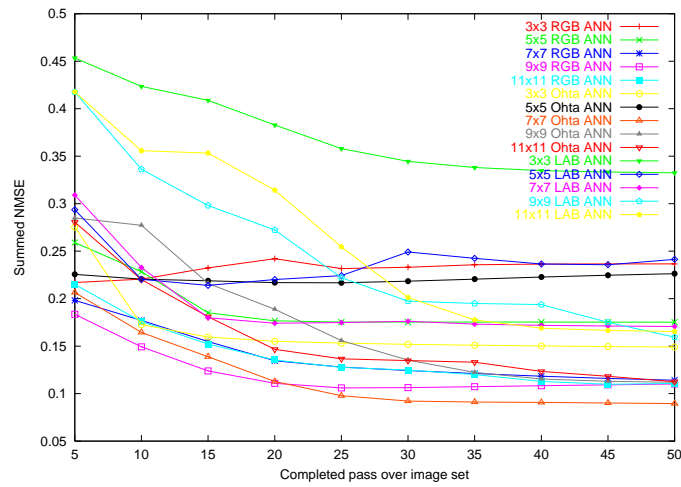


Figure 7.8. Testing error for 5% impulse noise, standard image set.

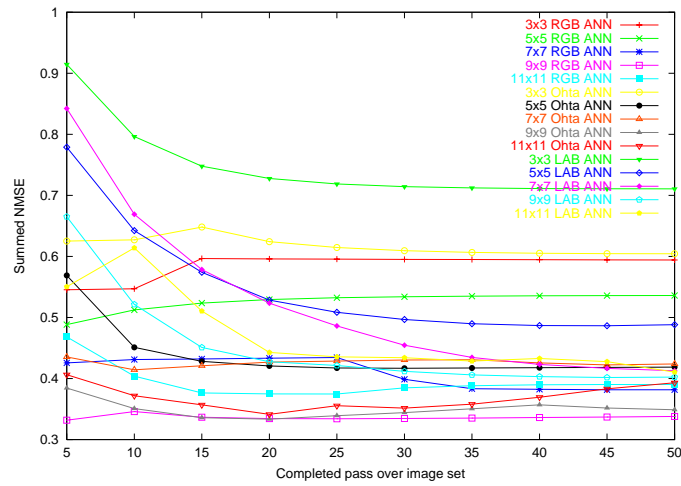


Figure 7.9. Testing error for  $\sigma^2=1000$  Gaussian noise, plastic component image set.

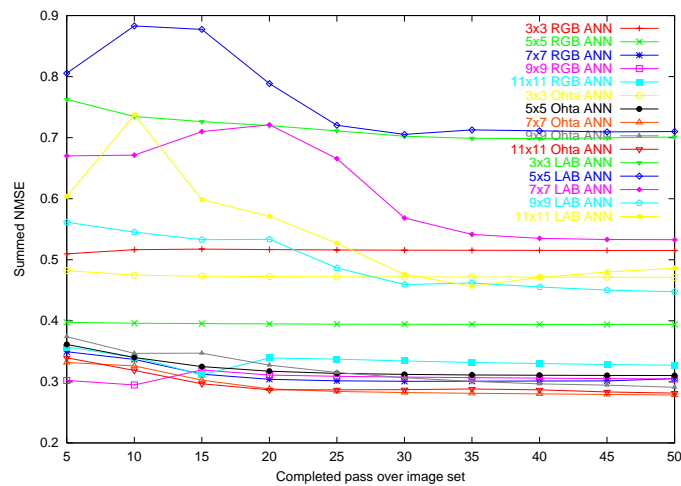


Figure 7.10. Testing error for 5% and  $\sigma^2=1000$  mixed noise, confectionery image set.

Compare this to the NMSE metric used for testing (Eq. 4.27) which is used when the images are converted back to their original RGB colourspace. This leads to the possibility that a relatively minor error in a given colourspace may give rise to a significantly larger error in the RGB colourspace after conversion, identified by the NMSE metric.

These observations are particularly relevant for the LAB colourspace for two reasons: firstly, the LAB space attempts to minimize perceptual errors so a small difference between LAB vectors can produce a considerable difference between RGB vectors.

Secondly, the neural network inputs and outputs are normalized, and the RGB gamut in LAB space occupies a wide dynamic range. However, typical RGB values will only occupy a restricted part of this range but the normalization factors must be constant to keep the input and output ranges between 0 and 1.

This leads to small neuron values which, due to limited floating point number support, can have significant truncation errors. It is mainly for this reason why the LAB space displayed seemingly inconsistent low training errors and high testing errors.

It is also interesting to note that the Ohta colourspace has greater success than RGB for images which have large variance, i.e. the standard images and images corrupted with the heaviest Gaussian noise; images with a dominant intensity component (the component and confectionery image sets contaminated with impulse or light Gaussian noise) are better suited to the RGB colourspace it would appear. This is almost certainly due to the impulse noise being added in the RGB gamut, leading to a more even distribution for that colourspace and thus easier to distinguish when analyzing an image correlated along the leading diagonal of the colour cube.

#### **7.4.4 COMPARISON WITH CONVENTIONAL FILTERS**

To evaluate the efficacy of using a neural network to remove noise from colour images, the saved networks which performed the best (listed in Tables 7.1 – 7.3) are compared with an appropriate conventional filter, selected from the best performing of the standard VMF, AMF or  $\alpha$ VMF with circular operator sizes between  $3 \times 3$  and  $11 \times 11$ . The comparison is done on an image-by-image basis, with the complete results listed in Appendix C3. A summary of the overall performance is given by calculating the average NMSE scores for the best ANN and conventional filters, shown in Tables 7.5 – 7.7.

##### **STANDARD IMAGES**

The results for the ANNs are broadly similar across all experiments: the ANN performs better on the training set than on the testing set, but the improvement over the conventional filter is only apparent on half the training images; Figure 7.11 shows the results from the 5% impulse noise experiment. The ANN has clearly had some difficulty in generalizing its function with the different classes of image, lending weight to the hypothesis that ANNs will have greater success at learning to remove noise from images with similar spatial and spectral characteristics. However, considering the overall view given by Table 7.5, the ANN does return considerably better noise figures than for the best conventional filter for all noise types.

##### **PLASTIC COMPONENT IMAGES**

With the exception of the moderate impulse noise experiment and the light mixed noise experiment the ANNs return better noise figures than the conventional filter for all the images in both training and testing sets (Figure 7.12 shows the heavy Gaussian noise experimental results). Table 7.6 shows that the ANN performs quite poorly in comparison to the standard VMF for the 5% impulse noise experiment, but

overall the ANN still outperforms the  $\alpha$ VMF for the light mixed noise experiment. Training the ANNs for this image set has clearly been successful for the great majority of the images, training and testing.

### **CONFECTIONERY IMAGES**

The experiments on the confectionery images are somewhat more mixed; the ANNs perform very poorly on both the impulse noise experiments and the light mixed noise experiment (Table 7.7). It is clear that the ANNs have learnt to remove Gaussian noise with great success in this image set, displaying much lower error scores than the best of the conventional filters but have difficulty in removing impulse noise. The problem of removing impulse noise is somewhat more difficult to specify, there is no definite rule in correctly classifying a pixel as noisy or uncorrupted; one only has to consider the success of non-linear filters versus linear filters in this regard.

Conversely, with Gaussian contamination, every pixel is affected to a lesser or greater extent; the ANN can quickly approximate the arithmetic mean filter and then specialize according to the training set statistics.

The lack of detailed features (such as sharp image contours) in this particular image set illustrates these characteristics. Figure 7.13 shows the results of the heavy mixed noise experiment.

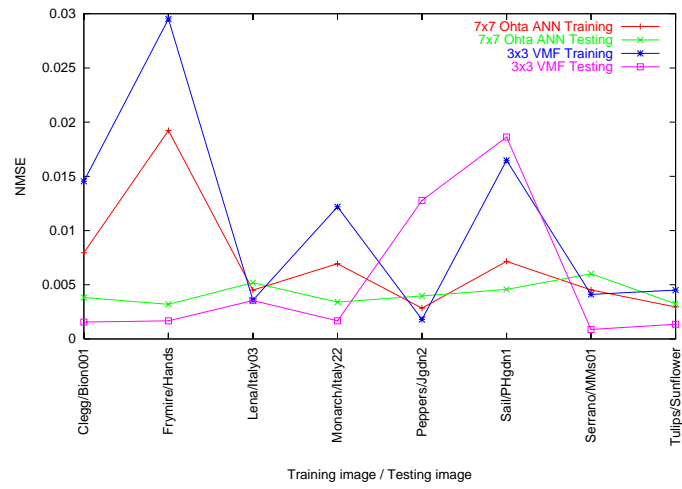


Figure 7.11. Filter comparison for 5% impulse noise, standard image set.

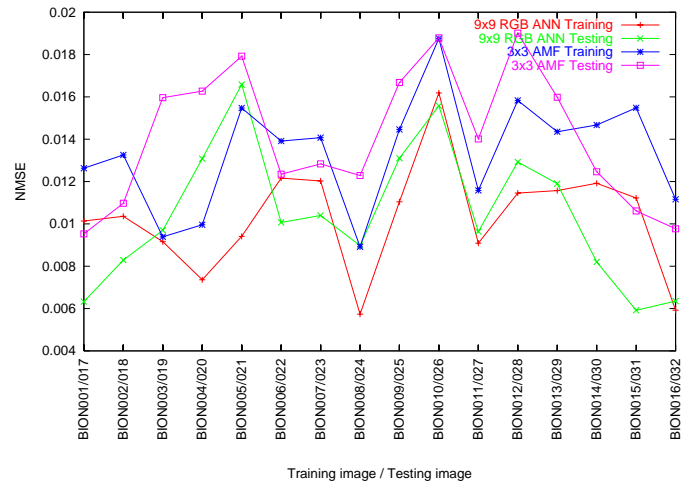


Figure 7.12. Filter comparison for  $\sigma^2=1000$  Gaussian noise, plastic component image set.

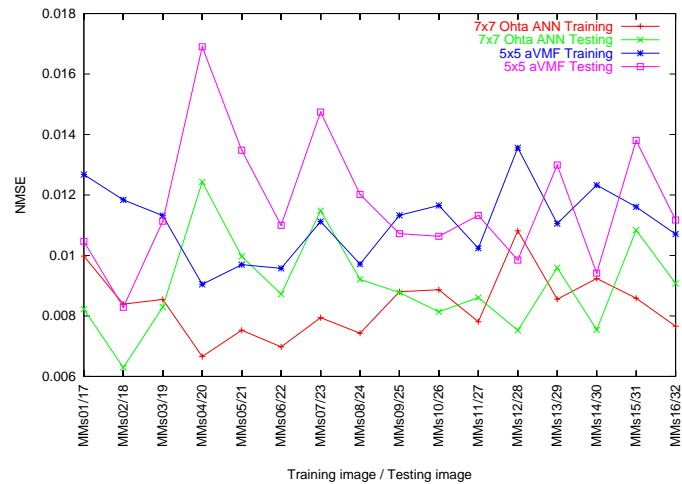


Figure 7.13. Filter comparison for 5% and  $\sigma^2=1000$  mixed noise, confectionery image set.

Noise type	Average NMSE		Conventional filter type
	Best performing ANN	Best conventional filter	
2% impulse	$3.635 \times 10^{-3}$	$7.467 \times 10^{-3}$	$3 \times 3$ VMF
5% impulse	$5.594 \times 10^{-3}$	$8.049 \times 10^{-3}$	$3 \times 3$ VMF
$\sigma^2=100$ Gaussian	$4.205 \times 10^{-3}$	$9.073 \times 10^{-3}$	$3 \times 3$ $\alpha$ VMF
$\sigma^2=1000$ Gaussian	$1.689 \times 10^{-2}$	$1.846 \times 10^{-2}$	$3 \times 3$ AMF
2% and $\sigma^2=100$ mixed	$7.528 \times 10^{-3}$	$9.302 \times 10^{-3}$	$3 \times 3$ $\alpha$ VMF
5% and $\sigma^2=1000$ mixed	$2.311 \times 10^{-2}$	$2.372 \times 10^{-2}$	$3 \times 3$ $\alpha$ VMF

**Table 7.5. ANN and conventional filter comparison for standard image set.**

Noise type	Average NMSE		Conventional filter type
	Best performing ANN	Best conventional filter	
2% impulse	$1.370 \times 10^{-3}$	$2.641 \times 10^{-3}$	$3 \times 3$ VMF
5% impulse	$3.502 \times 10^{-3}$	$3.174 \times 10^{-3}$	$3 \times 3$ VMF
$\sigma^2=100$ Gaussian	$2.339 \times 10^{-3}$	$4.490 \times 10^{-3}$	$3 \times 3$ $\alpha$ VMF
$\sigma^2=1000$ Gaussian	$1.037 \times 10^{-2}$	$1.373 \times 10^{-2}$	$3 \times 3$ AMF
2% and $\sigma^2=100$ mixed	$4.154 \times 10^{-3}$	$4.707 \times 10^{-3}$	$3 \times 3$ $\alpha$ VMF
5% and $\sigma^2=1000$ mixed	$1.436 \times 10^{-2}$	$1.860 \times 10^{-2}$	$3 \times 3$ $\alpha$ VMF

**Table 7.6. ANN and conventional filter comparison for plastic component image set.**

Noise type	Average NMSE		Conventional filter type
	Best performing ANN	Best conventional filter	
2% impulse	$1.796 \times 10^{-3}$	$8.757 \times 10^{-4}$	$3 \times 3$ VMF
5% impulse	$2.615 \times 10^{-3}$	$1.132 \times 10^{-3}$	$3 \times 3$ VMF
$\sigma^2=100$ Gaussian	$1.670 \times 10^{-3}$	$2.637 \times 10^{-3}$	$3 \times 3$ $\alpha$ VMF
$\sigma^2=1000$ Gaussian	$6.239 \times 10^{-3}$	$9.636 \times 10^{-3}$	$5 \times 5$ AMF
2% and $\sigma^2=100$ mixed	$3.133 \times 10^{-3}$	$2.762 \times 10^{-3}$	$3 \times 3$ $\alpha$ VMF
5% and $\sigma^2=1000$ mixed	$8.703 \times 10^{-3}$	$1.142 \times 10^{-2}$	$5 \times 5$ $\alpha$ VMF

**Table 7.7. ANN and conventional filter comparison for confectionery image set.**

#### 7.4.5 COMBINING NEURAL NETWORK WITH NEAREST NEIGHBOUR SELECTION

It is clear from the observations above that the ANNs response to Gaussian contaminated images is far better than to the impulse noise contaminated images. The reason for this is that although the vast majority of pixels in the image are noise-free, the ANNs distort all pixels to some extent, noisy or not, increasing the NMSE score.

A method for improving this has been developed by adding a further ‘nearest neighbour’ module after the ANN.

After the ANN has produced a tristimulus output, this value is fed to a ‘nearest neighbour’ processing stage which determines which of the original local window pixels is closest to the ANN’s output and uses this as the replacement pixel value: this has the effect of eliminating noisy pixels according to the ANN’s rationale whilst still selecting a pixel of colour and intensity present in the original distribution in noise-free areas. Note that this will only work on noise types in which the majority of pixels are left unchanged.

The result of applying the ‘nearest neighbour’ module to the impulse noise experiments is given in Tables 7.8 – 7.10.



It can be seen that the ‘nearest neighbour’ method drastically improves the error score in all cases, especially for those with the highest errors. Although this method does not improve the error scores for the confectionery image set to surpass the conventional filter, it does significantly improve over the VMF for the moderate noise case for the confectionery image set.

	Average NMSE		
Noise type	Standard ANN	ANN with ‘Nearest Neighbour’	Best conventional filter (3 × 3 VMF)
2% impulse	$3.635 \times 10^{-3}$	$3.102 \times 10^{-3}$	$7.467 \times 10^{-3}$
5% impulse	$5.594 \times 10^{-3}$	$4.578 \times 10^{-3}$	$8.049 \times 10^{-3}$

**Table 7.8. ANN with ‘nearest neighbour’ and conventional filter comparison for standard image set.**

	Average NMSE		
Noise type	Standard ANN	ANN with ‘Nearest Neighbour’	Best conventional filter (3 × 3 VMF)
2% impulse	$1.370 \times 10^{-3}$	$1.257 \times 10^{-3}$	$2.641 \times 10^{-3}$
5% impulse	$3.502 \times 10^{-3}$	$2.843 \times 10^{-3}$	$3.174 \times 10^{-3}$

**Table 7.9. ANN with ‘nearest neighbour’ and conventional filter comparison for plastic component image set.**

	Average NMSE		
Noise type	Standard ANN	ANN with ‘Nearest Neighbour’	Best conventional filter (3 × 3 VMF)
2% impulse	$1.796 \times 10^{-3}$	$1.291 \times 10^{-3}$	$8.757 \times 10^{-4}$
5% impulse	$2.615 \times 10^{-3}$	$1.898 \times 10^{-3}$	$1.132 \times 10^{-3}$

**Table 7.10. ANN with ‘nearest neighbour’ and conventional filter comparison for confectionery image set.**

## 7.5 CONCLUDING REMARKS

This chapter has investigated the applicability of neural networks to removing noise of different types, with the hypothesis that the neural network would train better on images exhibiting similar spatial and chromatic characteristics, as opposed to images of a distinct nature; this is borne out by the consistency of the results for the plastic component and confectionery image sets, as opposed to the somewhat more disparate performance on the standard image set, especially on the testing set.

In general, the ANNs showed a significant improvement over the use of conventional filters, although in a qualitative sense this is to a large extent due to the retention of detail rather than superior suppression of noise, as shown in the examples of ANN processed images in Figure 7.14a-d – Figure 7.16a-d. However, as Tables 7.5 – 7.7 have shown, the overall performance gain against the conventional filters tested is plain to see.

In summary, the achievements of this work are as follows:

- The application of ANNs to the problem of noise removal has been carried out, verifying previous researchers' work (see Davies and Greenhill<sup>142</sup>).
- Analysis of ANNs over a larger image set has demonstrated that the effectiveness of ANNs at this task is not a spurious result.
- Their work has been extended by investigating different noise types and their combinations, along with different operator sizes.
- The suitability of using ANNs in the removal of noise in colour images has been researched and proven, with discussion and analysis of the ANNs behaviour for this application.
- The suitability of alternative colourspace has been investigated and proven.
- The performance of the ANNs has been compared with the best of the conventional filters discussed in Chapter 4 and demonstrated to be superior.

- A method to improve the retention of the original image characteristics after impulse noise filtering has been developed and its efficacy proven.

In closing, the suitability of ANNs at removing noise from colour images has been amply demonstrated and shown to rival well-known conventional filters; the aims stated at the beginning of the chapter have been investigated and the hypotheses verified; namely, that:

- MLPs are better suited to removing noise consistently from related rather than disparate images.
- Larger windows improve performance.
- Training MLPs represent a viable alternative to traditional filter design provided sufficient training can be done.

Indeed, as many filters require  $O(n^2)$  or  $O(n^3)$  operations, and the network (once trained) requires  $O(n)$  operations, the network's computational load for large windows is similar to that for a traditional filter utilizing a smaller window.

It was also demonstrated that removing noise using ANNs was achievable within reasonable training time.

From these foundations, using ANNs for colour image enhancement, restoration or noise removal processes which would otherwise be difficult to model conventionally is a viable proposition.



Figure 7.14a. Original 'Monarch' image.



Figure 7.14b. 'Monarch' image with  $\sigma^2=1000$  Gaussian noise.

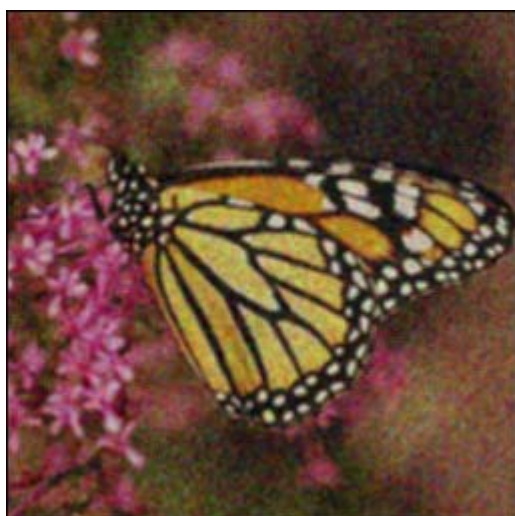


Figure 7.14c. Figure 7.14b filtered by  $3 \times 3$  AMF.



Figure 7.14d. Figure 7.14b filtered by  $5 \times 5$  Ohta ANN.



**Figure 7.15a.** Original plastic component image.



**Figure 7.15b.** Plastic component image with 5% impulse noise.



**Figure 7.15c.** Figure 7.15b filtered by  $3 \times 3$  VMF.



**Figure 7.15d.** Figure 7.15b filtered by  $11 \times 11$  RGB ANN.





Figure 7.16a. Original confectionery image.

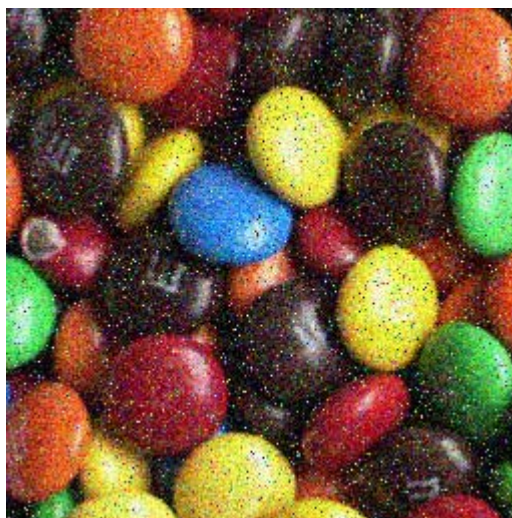


Figure 7.16b. Confectionery image with mixed 5% impulse and  $\sigma^2=1000$  Gaussian noise.



Figure 7.16c. Figure 7.16b filtered by  $5 \times 5$   $\alpha$ VMF.



Figure 7.16d. Figure 7.16b filtered by  $9 \times 9$  Ohta ANN.

# Chapter 8: Conclusions

This thesis has examined a range of issues pertinent to the restoration or enhancement of digital colour images. The first three chapters provide the necessary background information for the reader, the fourth gives a much needed overview and discussion of the performance of contemporary filters, and the final three chapters extended previous work and developed some original ideas with respect to filtering colour images.

## 8.1 REVIEW OF THESIS

Chapter 1 introduced the subject of digital image processing and the recent field of colour image processing. Colour was defined by discussing the phenomenon of human colour vision, and an outline given of the necessity of using colour machine vision techniques for a range of applications. The requirement for effective colour filtering techniques was then described in terms of the characteristics of colour image capture equipment, data compression and encoding techniques, display restrictions and transmission abnormalities.

Chapter 2 discussed popular methods of representing colour. Their origins were considered, along with their relative merits, shortfalls and applications. Methods for converting images between the various representations were also given.

Chapter 3 described the framework necessary for developing and measuring colour image processing algorithms. The requirements were discussed and basic details of the implementation given. The reasoning behind the selection of images used in the thesis was summarized according to their characteristics and prevalence in the published literature.

Chapter 4 introduced the concept of the local window operator and its association with convolution operators. Basic local window methods of noise removal in

greyscale images were described before outlining the difficulties of applying these operators in a component-wise fashion to multivariate images. The reader was introduced to the now well-known vector median filter before embarking on a description of alternative, oft-touted ‘superior’ vector filters. Prior to comparing the efficacy of these filters, methods for measuring performance were detailed, both for greyscale and colour application, and their merits discussed.

A number of experiments were performed on a range of images contaminated with different noise types, leading to the conclusion that the various filters had merits and shortcomings that depended greatly on the statistics of the local window on which they were used. The achievements of this work may be summarized as:

- The published capabilities of established filters were challenged by subjecting them to rigorous testing.
- The analysis and discussion of the filter characteristics led to the development of extensions to these filters.
- Differences in the definitions of the MCRE were discovered and led to a new definition for the MCRE, although qualitative measurements are yet to be carried out (Section 4.6).
- It was noted that the metrics themselves differed in the manner in which they ranked the filters, prompting the reader to be cautious when presented with quantitative performance data.

A brief introduction to artificial neural networks was also given to provide background information for Chapter 7.

Chapter 5 developed and analyzed a number of original extensions based on the vector median filter. Briefly, the extensions comprised:

- An investigation into the applicability of using alternative colourspaces.
- A hybrid mean-median filter based on the  $\alpha$ VMF to cope with unknown or mixed noise.
- A switching rank order filter to retain detail when removing impulse noise.



- A novel weighting scheme based on observation of the statistics of natural images.
- An extension to reduce computational overhead for the distance calculations often necessary in vector filtering.

Each of these extensions was found to endow greater performance in one or more areas; in reduced computational load or improved retention of detail, for example.

Chapter 6 presented an improvement on a somewhat more specialized filter: Davies' truncated median filter (commonly known as the 'mode filter'), well-known for its image enhancement properties, was investigated and a novel method developed for use on multivariate images. Although highlighting the difficulty in transferring scalar techniques into vector domains, the new method was shown to reduce to the action of Davies' filter for greyscale images, and further work performed relating to its noise-rejection properties in addition to those of enhancement. For heavy contamination, it was proven that the new filter is quite superior to the vector median filter.

Finally, quantitative confirmation of the theoretical work of Davies on the subject of edge-shifting was carried out and the model modified to take into account the digitized nature of images.

Chapter 7 investigated the efficacy of filters which *learned* the noisy characteristics of colour images to subsequently remove or suppress the noise. Previous work was described and extended to colour images. The same noise types used in the comprehensive comparison in Chapter 4 were employed here; as a result important observations regarding the behaviour of ANNs could be made, one of which led to the development of an improvement in removing impulse noise. The merits and shortcomings of ANNs as colour noise-removing filters were discussed, but most importantly, the applicability of using ANNs for such a task was clearly shown.

## 8.2 FUTURE WORK

From the work in Chapter 4, it is clear that none of the popular metrics give a full account of a filter's performance. We would like to see consideration given to the development of a measurement score which, as well as ranking overall distortion, will measure the edge-shift, chromatic distortion and perceived quality to provide a more balanced view of a filter's properties. The last item is difficult to rank due to the non-linear characteristics of the human visual system, although it would be interesting to see if ANNs could be trained with consistent results to match human-based opinion scores.

We have also observed the variation of a filter's performance according to local statistics; quite recently, a number of switching techniques have been suggested in the literature based on local and global statistics; it would be instructive to see how well the proposed extensions integrate into such schemes, or indeed whether they may form part of the analysis of an image's properties. This is especially true for the work done on the ANN's: small networks could be specifically trained for a great number of situations with selection of the best ANNs made by a decision-tree type algorithm.

The properties of Davies' truncated median filter made it ideal both as a pre- and post-processing step for different types of segmentation algorithm. We should like to investigate whether this is the case for colour segmentation techniques and quantify its worth.

The work on the effectiveness of ANNs at removing noise should also be continued: detailed weight analysis may lead to more computationally efficient 'fixed' non-linear or linear filters and the edge distortion should be quantified and isotropic action verified. Although each ANN was trained for a specific noise type and image set, the performance of each was not cross-checked; although chromatic aberrations were demonstrated early in Chapter 7, this does not preclude the possibility of a 'general' filter that one may employ on an unknown image. The distortions

introduced by ANNs should also be analyzed; a suitable starting point may be the measurement of edge-shifting, similar to that carried out at the end of Chapter 6.

Finally, only the use of multi-layer perceptrons was investigated; many other types of network exist and may well be better suited to the problem of noise removal and merit further investigation.

# Appendix A

## A1 CIE DEFINITIONS

Due to the perceptual nature of colour, the CIE (Commission Internationale de l'Éclairage – the International Commission on Illumination) have introduced strict definitions describing colour sensations with respect to physically measurable quantities<sup>†</sup>:

**Brightness** *Brightness* is defined by the CIE as *the attribute of a visual sensation according to which an area appears to exhibit more or less light.*

**Luminance** Luminance is brightness weighted by a spectral sensitivity curve characteristic of human vision, peaking at 555nm. When a spectral power density function is integrated using this curve the result is *CIE Luminance*, denoted  $Y$ . To compute absolute luminance from linear RGB the following equation from Rec. 709<sup>50</sup> is used:

$$Y = 0.2125R + 0.7154G + 0.0721B$$

**Lightness** The human eye has a roughly logarithmic response to brightness changes, much as the ear does to sound pressure changes; a light source five times the luminance of a reference source only appears twice as bright. This non-linear perception of luminance is *CIE Lightness*, denoted  $L^*$  and defined as the modified cube root of  $Y$ :

$$L^* = 116 \left( \frac{Y}{Y_n} \right)^{\frac{1}{3}} - 16; \quad 0.008856 < \frac{Y}{Y_n} < 1 \quad 0 < L^* < 100$$

---

<sup>†</sup> Hunt, R.W.G (1991) *Measuring Colour*, Ellis Horwood Ltd., Chichester.

$$L^* = 903.3 \left( \frac{Y}{Y_n} \right); \quad 0.008856 \geq \frac{Y}{Y_n} \quad 0 < L^* < 100$$

where  $Y_n$  is the luminance of the white reference and  $(Y/Y_n)$  is scaled from 0 to 1. The asterisk suffix denotes that the quantity has undergone a transformation.

- Hue** The CIE states that *hue is the attribute of a visual sensation according to which an area appears to be similar to one of the perceived colours, red, yellow, green and blue, or a combination of two of them*. In other words, a change in hue gives rise to a change in the colour perceived.
- Saturation** *Saturation is the colourfulness of an area judged in proportion to its brightness*, i.e. saturation is a measure of the purity of the perceived colour. A fully desaturated hue exhibits no colour; only the luminance component contributes to its output, i.e. it will be purely greyscale.
- Red** Pure red is defined as 700nm according to Rec. 709.
- Green** Pure green is defined as 546.1nm according to Rec. 709.
- Blue** Pure blue is defined as 435.8nm according to Rec. 709.

## A2 TELEVISION COLOUR SPACES

European (PAL)  $Y'U'V'$ , American (NTSC)  $Y'I'Q$  and other associated terrestrial broadcast standards are analog, non-linear systems which, although not strictly related to digital image coding/processing are worth mentioning because a number of workers use television luminance and chrominance relations to derive transforms to convert from RGB to other colour spaces suitable for their applications.

Broadcast systems differ in that they are *gamma-corrected*. Voltage applied to the cathode ray tube is not proportional to intensity, so the transmitted signal is pre-corrected to compensate for this by a transfer function with a power law denoted by

gamma. For accurate colour calculations, this pre-correction must be undone to return to linear signals. Signals that are gamma-corrected are usually primed.

European television (PAL and SECAM) uses  $Y'U'V'$  components, and whilst the symbols denote similar types of quantity ( $Y'$  for luminance,  $U'$  and  $V'$  for chrominance) they are not related to CIE standards.  $Y'$  is actually referred to as *luma*, to prevent confusion with the CIE definition.  $U'$  and  $V'$  are bipolar signals, and therefore assume negative as well as positive values.

To convert from  $Y'U'V'$  to gamma-corrected  $R'G'B'$ , the equations are:

$$\begin{aligned} Y' &= 0.299R' + 0.587G' + 0.114B' \\ U' &= 0.493(B' - Y') = -0.147R' - 0.289G' + 0.436B' \\ V' &= 0.877(R' - Y') = 0.615R' - 0.515G' - 0.100B' \end{aligned}$$

And:

$$\begin{aligned} R' &= Y' + 1.140V' \\ G' &= Y' - 0.396U' - 0.581V' \\ B' &= Y' + 2.029U' \end{aligned}$$

The gamma law is assumed to be 2.8.

American NTSC coded television uses  $Y'I'Q'$  components, which like PAL, represent one luminance- and two chrominance-coded channels, where  $I'$  is the inphase chrominance and  $Q'$  is the quadrature chrominance.  $I'$  and  $Q'$  are derived by rotating the  $U'V'$  vector by 33 degrees. The gamma law for NTSC is different at 2.2.

Again, to convert from  $R'G'B'$  to  $Y'I'Q'$  and vice versa, the equations are:

$$\begin{aligned} Y' &= 0.299R' + 0.587G' + 0.114B' \\ I' &= -0.27(B' - Y') + 0.74(R' - Y') = 0.596R' - 0.274G' + 0.322B' \\ Q' &= 0.41(B' - Y') + 0.48(R' - Y') = 0.212R' - 0.523G' - 0.311B' \end{aligned}$$

And:

$$R' = Y' + 0.956I' + 0.621Q'$$

$$G' = Y' - 0.272I' - 0.647Q'$$

$$B' = Y' - 1.105I' + 1.702Q'$$

When the NTSC standard was ratified in 1953, the chromaticity co-ordinates of the transmission primaries were quite different to those of the EBU (European Broadcasting Union), mainly because of the CRT phosphors in use at the time. Since then, the NTSC system has changed its primaries many times, and they are now very similar to those of the EBU, making it possible to convert between one system and the other. To achieve this, the following relations are used:

$$I' = -\frac{0.27}{0.493}U' + \frac{0.74}{0.877}V'$$

$$Q' = -\frac{0.41}{0.493}U' + \frac{0.48}{0.877}V'$$

This can be simplified to:

$$I' = -0.547U' + 0.843V'$$

$$Q' = 0.831U' + 0.547V'$$

with the reverse relation:

$$U' = -0.547I' + 0.843Q'$$

$$V' = 0.831I' + 0.547Q'$$

## A3 CONVERSION BETWEEN RGB STANDARDS

To transform from SMPTE 240M (or SMPTE RP-145) RGB primaries to Rec. 709 the following matrix is used:<sup>†</sup>

$$\begin{bmatrix} R_{709} \\ G_{709} \\ B_{709} \end{bmatrix} = \begin{bmatrix} 0.939555 & 0.050173 & 0.010272 \\ 0.017775 & 0.965795 & 0.016430 \\ -0.001622 & -0.004371 & 1.005993 \end{bmatrix} \begin{bmatrix} R_{145} \\ G_{145} \\ B_{145} \end{bmatrix}$$

and for EBU 3213 primaries the following relation holds:

$$\begin{bmatrix} R_{709} \\ G_{709} \\ B_{709} \end{bmatrix} = \begin{bmatrix} 1.044036 & -0.044036 & 0 \\ 0 & 1 & 0 \\ 0 & 0.011797 & 0.988203 \end{bmatrix} \begin{bmatrix} R_{EBU} \\ G_{EBU} \\ B_{EBU} \end{bmatrix}$$

It should be realized that transforming between different RGB systems may lead to out-of-gamut values.

---

<sup>†</sup> Benson, K.B. (1992) *Television Engineering Handbook feat. HDTV Systems*, McGraw-Hill., New York.



# Appendix B

Please browse the attached CD-ROM for program code and an electronic version of this thesis.

# Appendix C

This appendix contains the full results of the training and testing of the neural networks used for noise suppression in Chapter 7.

## C1 ANN TRAINING RESULTS

For each noise type, the total squared error over all the training images in each pass is shown in Figures C1.1a-f for the standard image set, Figures C1.2a-f for the plastic component image set and Figures C1.3a-f for the confectionery image set.

## C2 ANN TESTING RESULTS

Testing was conducted by converting each noisy image to the appropriate colourspace, applying the neural network, then converting back to RGB and obtaining a NMSE measure. A second set of images for each image set was used to confirm each network's ability to generalize within an image class.

During training, the state of each network was saved every five passes over each image set; this ensures that during testing the 'best' set of weights can be identified.

For each noise type, the NMSE summed over all the training and testing images for each saved network is shown in Figures C2.1a-f for the standard image set, Figures C2.2a-f for the plastic component image set and Figures C2.3a-f for the confectionery image set.

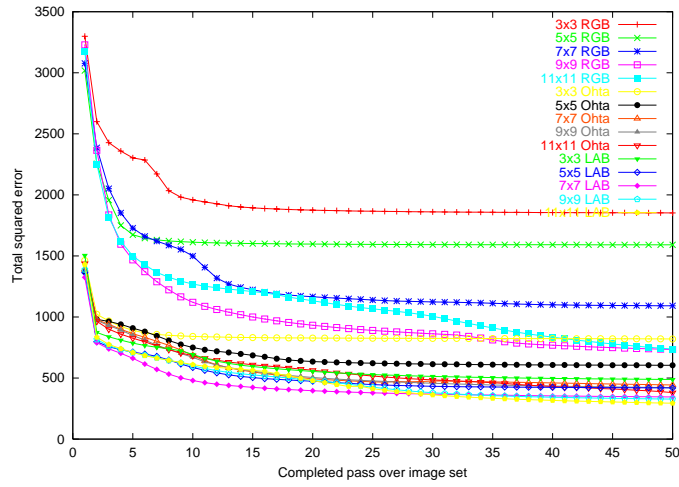


Figure C1.1a. Training error for 2% impulse noise, standard image set.

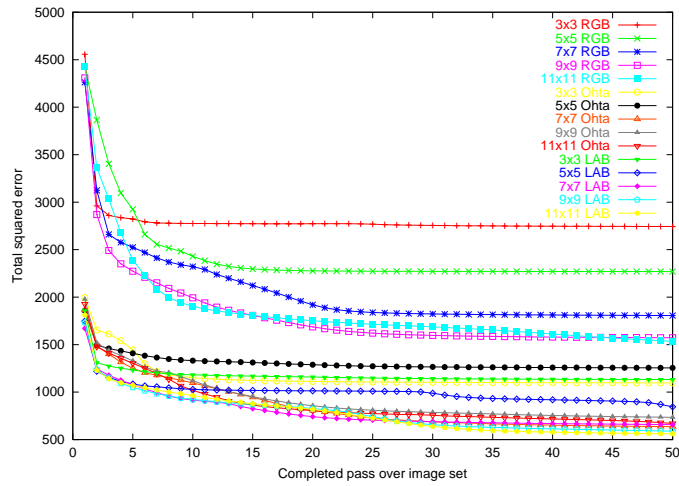


Figure C1.1b. Training error for 5% impulse noise, standard image set.

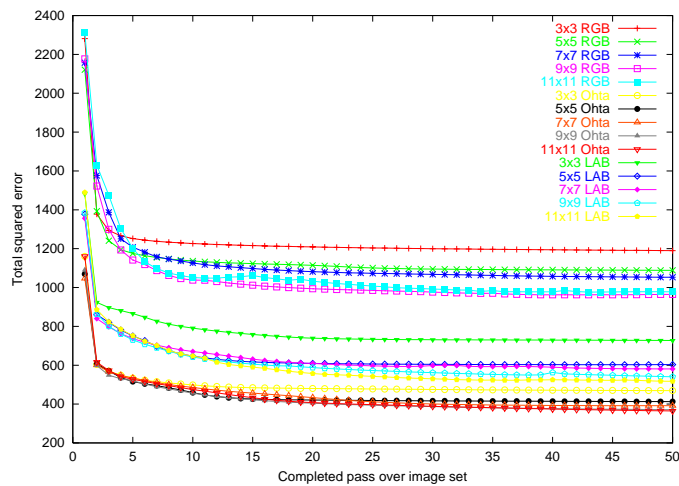


Figure C1.1c. Training error for  $\sigma^2=100$  Gaussian noise, standard image set.

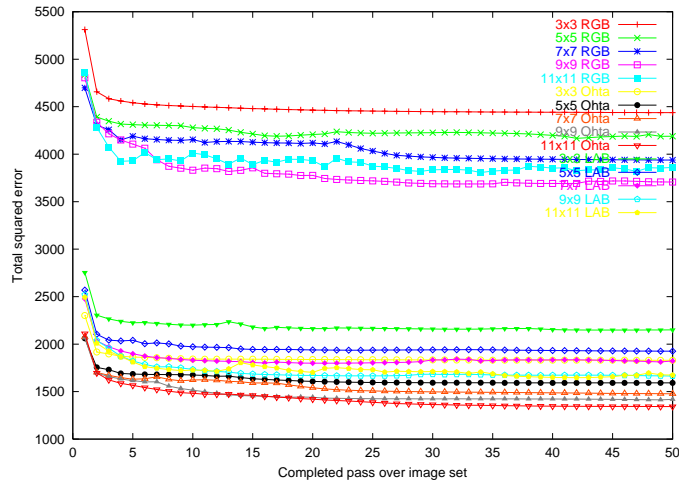


Figure C1.1d. Training error for  $\sigma^2=1000$  Gaussian noise, standard image set.

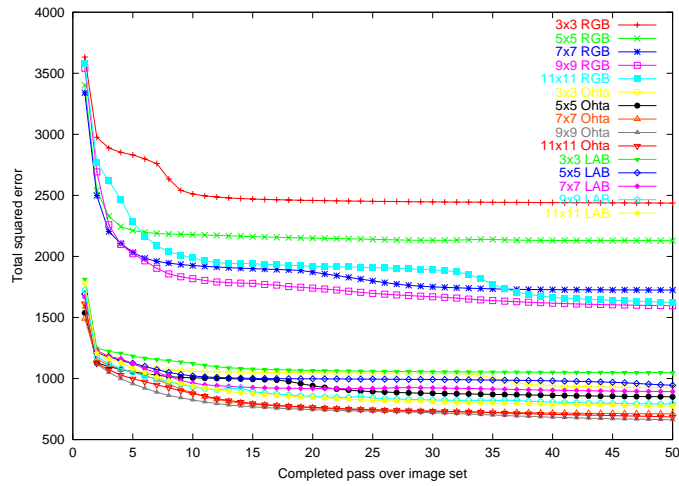


Figure C1.1e. Training error for 2% and  $\sigma^2=100$  mixed noise, standard image set.

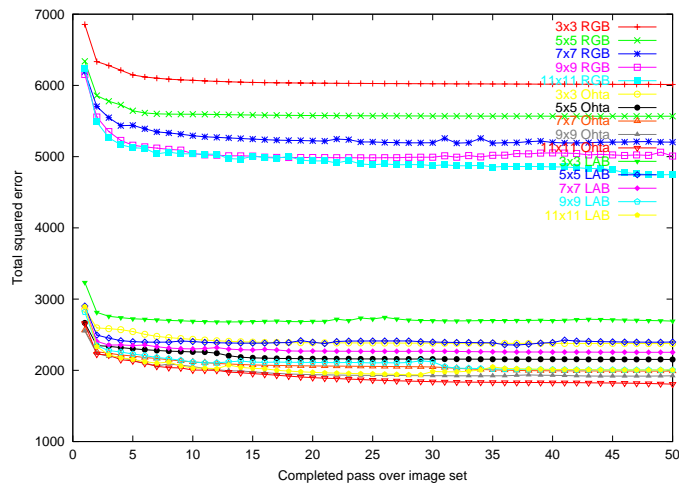
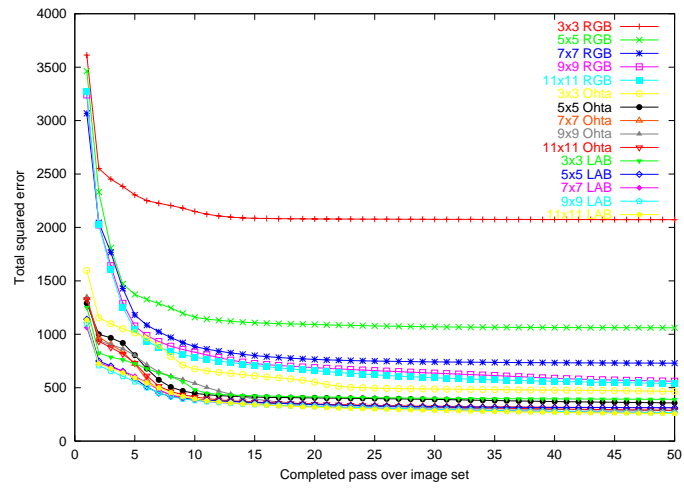
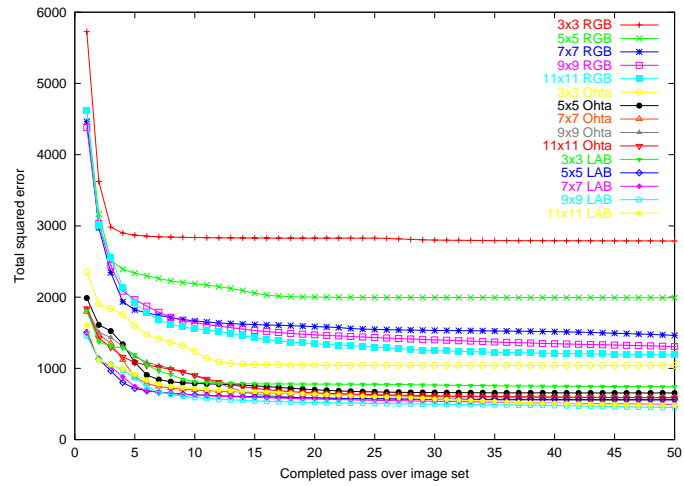


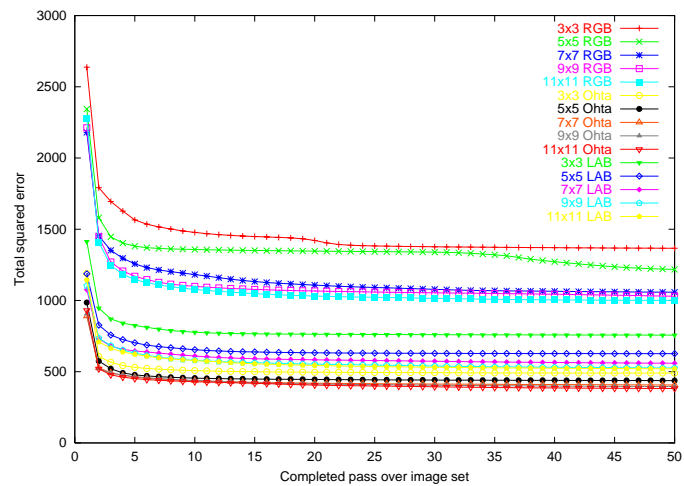
Figure C1.1f. Training error for 5% and  $\sigma^2=1000$  mixed noise, standard image set.



**Figure C1.2a. Training error for 2% impulse noise, plastic component image set.**



**Figure C1.2b. Training error for 5% impulse noise, plastic component image set.**



**Figure C1.2c. Training error for  $\sigma^2=100$  Gaussian noise, plastic component image set.**

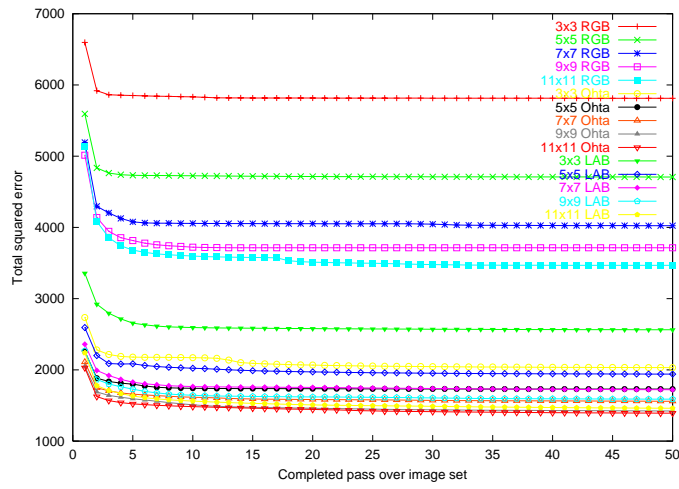


Figure C1.2d. Training error for  $\sigma^2=1000$  Gaussian noise, plastic component image set.

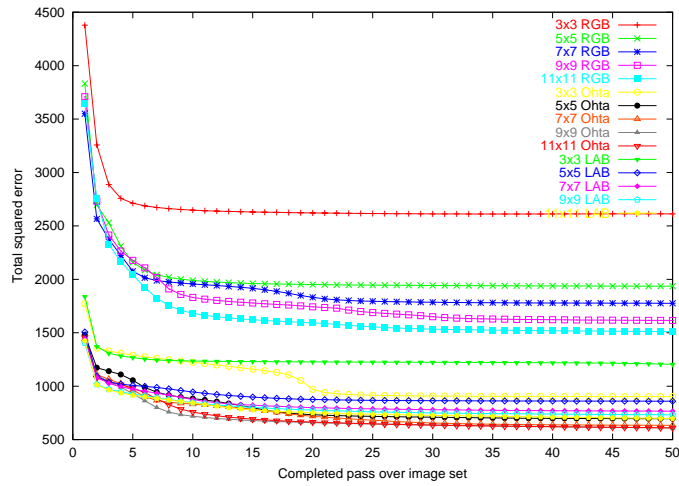


Figure C1.2e. Training error for 2% and  $\sigma^2=100$  mixed noise, plastic component image set.

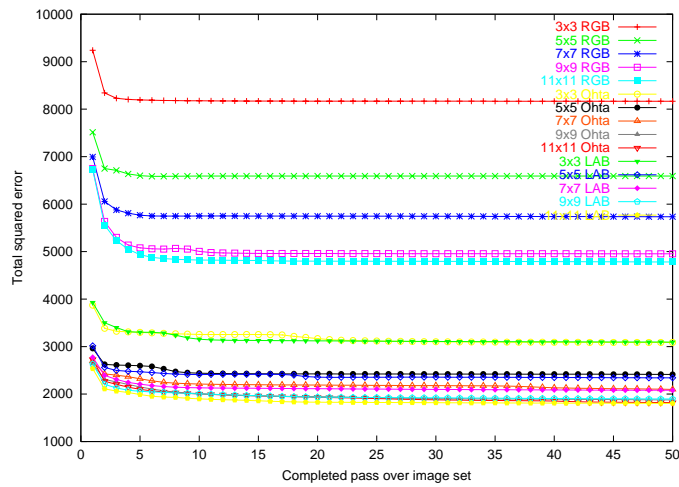


Figure C1.2f. Training error for 5% and  $\sigma^2=1000$  mixed noise, plastic component image set.

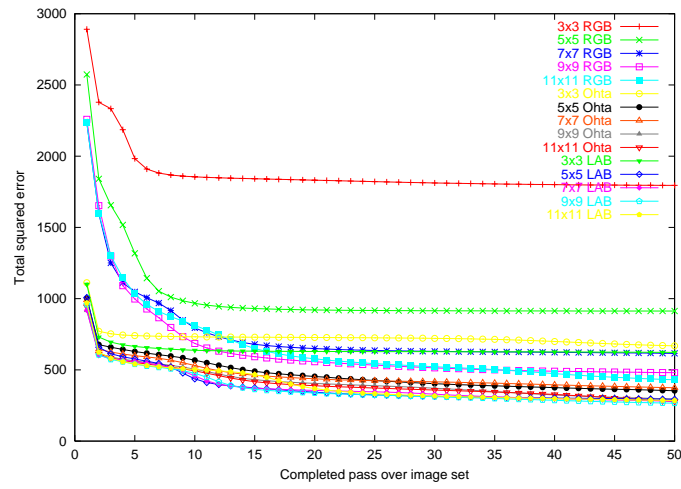


Figure C1.3a. Training error for 2% impulse noise, confectionery image set.

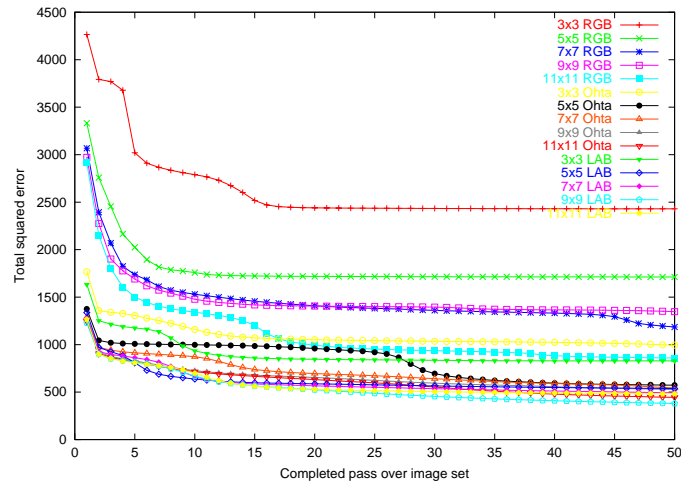


Figure C1.3b. Training error for 5% impulse noise, confectionery image set.

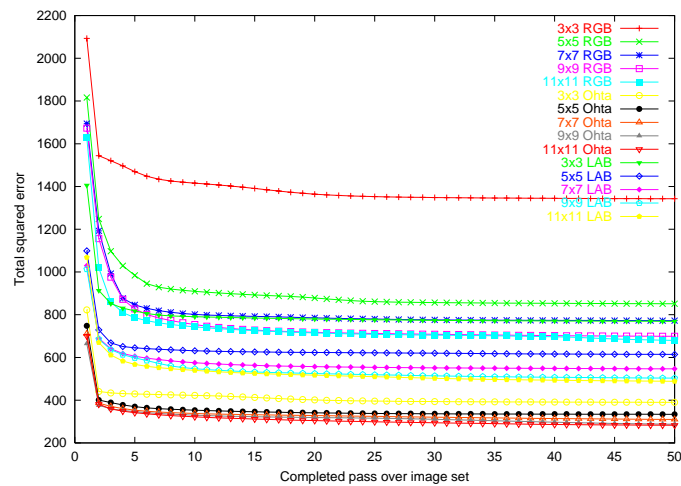


Figure C1.3c. Training error for  $\sigma^2=100$  Gaussian noise, confectionery image set.

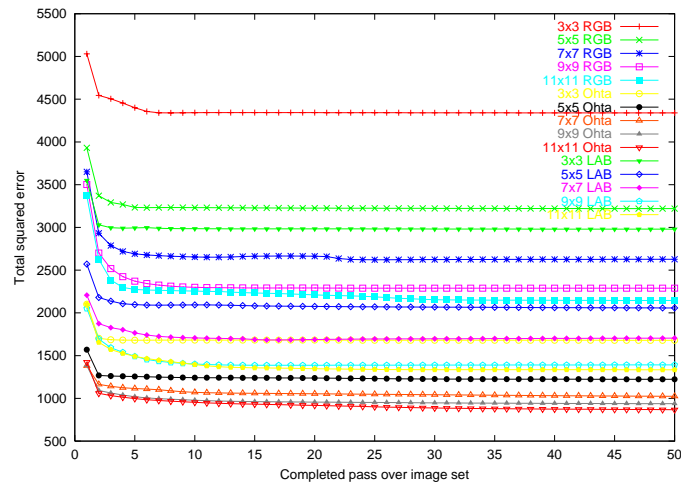


Figure C1.3d. Training error for  $\sigma^2=1000$  Gaussian noise, confectionery image set.

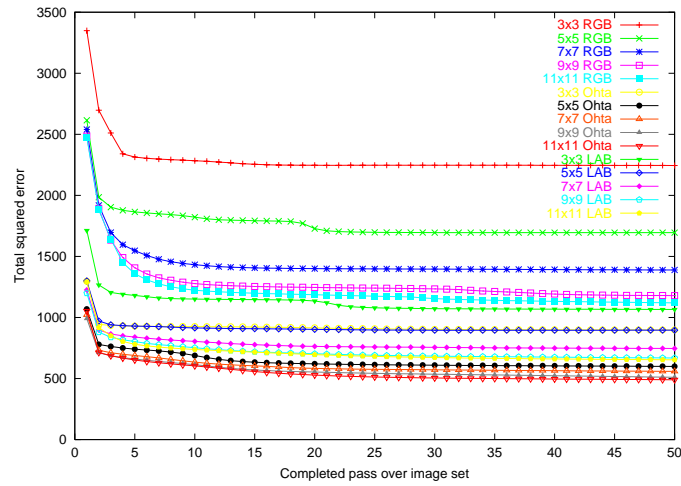


Figure C1.3e. Training error for 2% and  $\sigma^2=100$  mixed noise, confectionery image set.

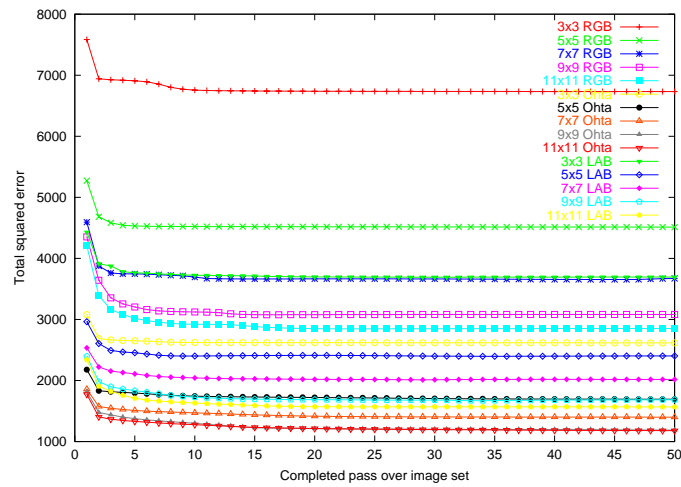


Figure C1.3f. Training error for 5% and  $\sigma^2=1000$  mixed noise, confectionery image set.



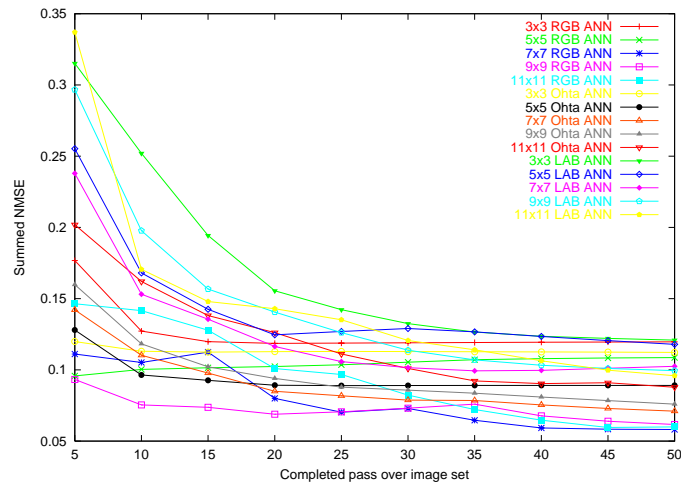


Figure C2.1a. Testing error for 2% impulse noise, standard image set.

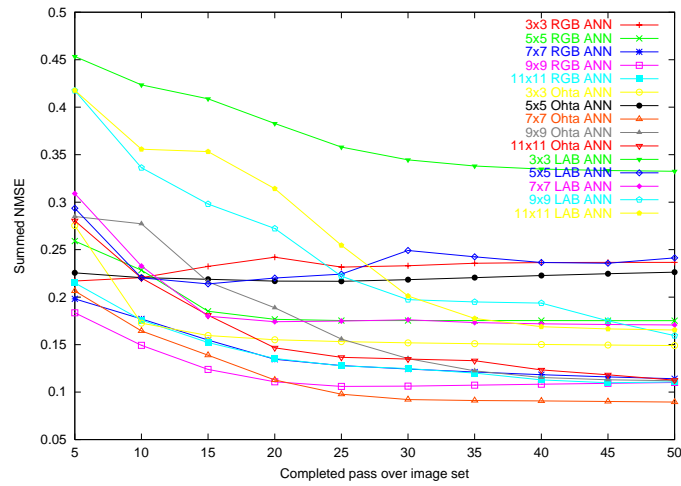


Figure C2.1b. Testing error for 5% impulse noise, standard image set.

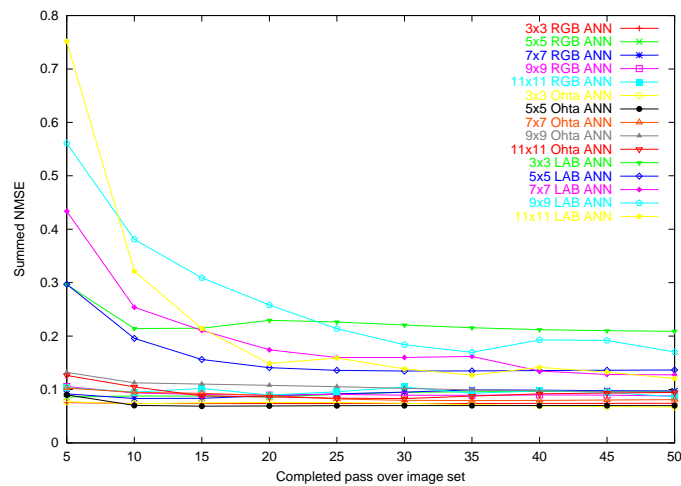


Figure C2.1c. Testing error for  $\sigma^2=100$  Gaussian noise, standard image set.

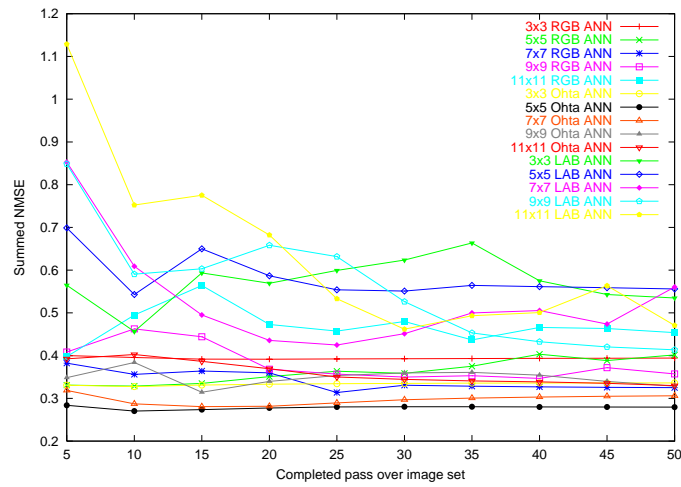


Figure C2.1d. Testing error for  $\sigma^2=1000$  Gaussian noise, standard image set.

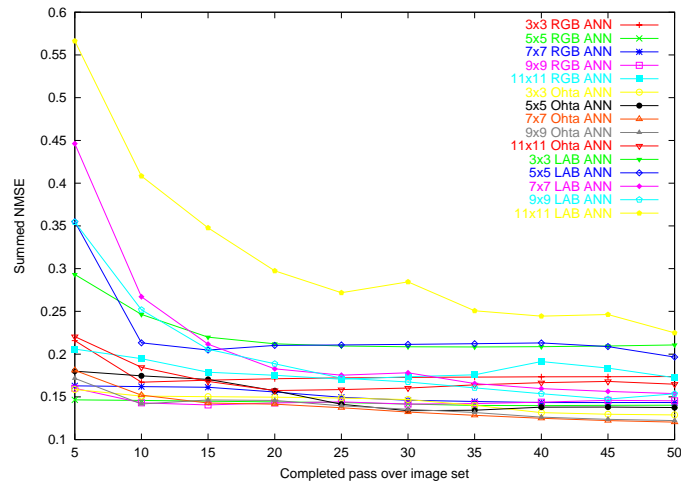


Figure C2.1e. Testing error for 2% and  $\sigma^2=100$  mixed noise, standard image set.

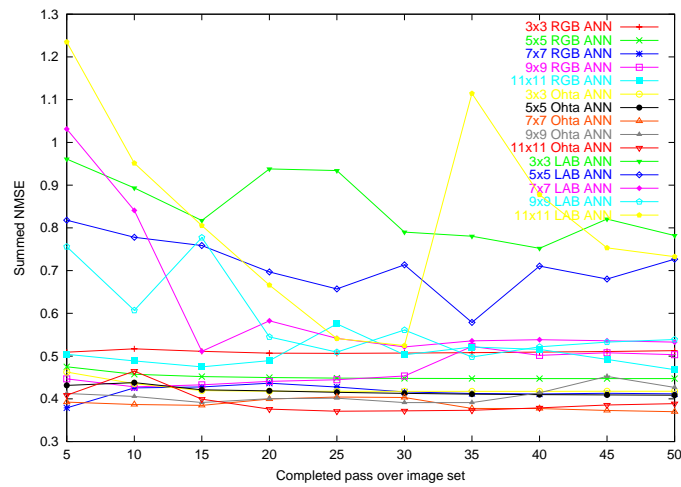


Figure C2.1f. Testing error for 5% and  $\sigma^2=1000$  mixed noise, standard image set.

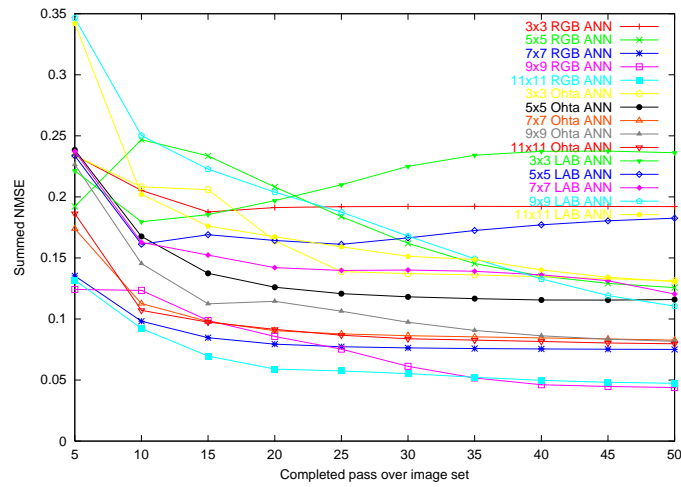


Figure C2.2a. Testing error for 2% impulse noise, plastic component image set.

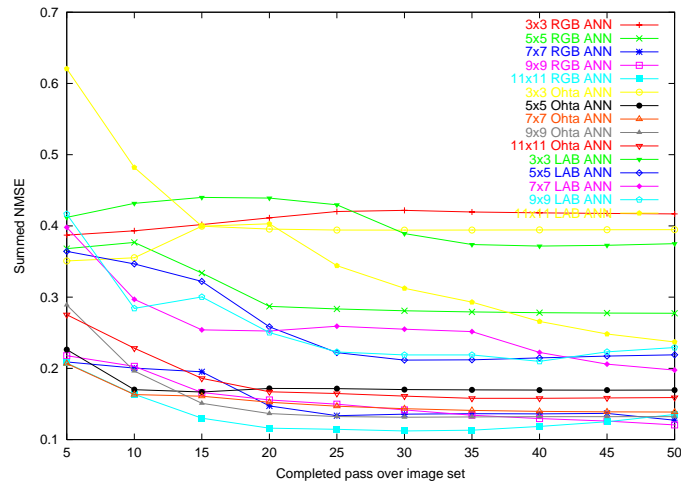


Figure C2.2b. Testing error for 5% impulse noise, plastic component image set.

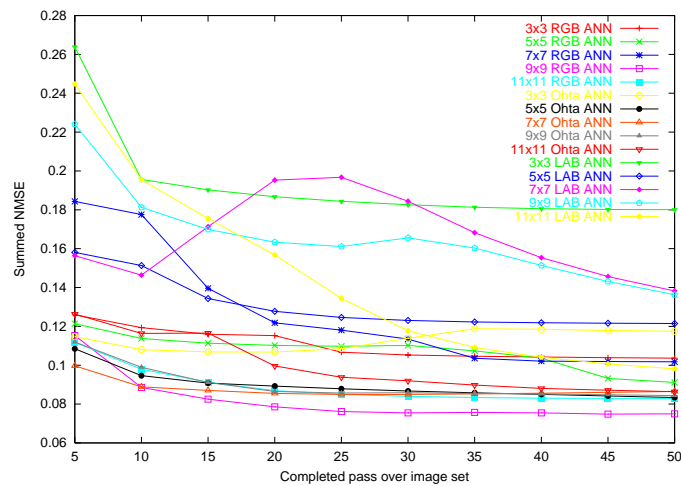


Figure C2.2c. Testing error for  $\sigma^2=100$  Gaussian noise, plastic component image set.

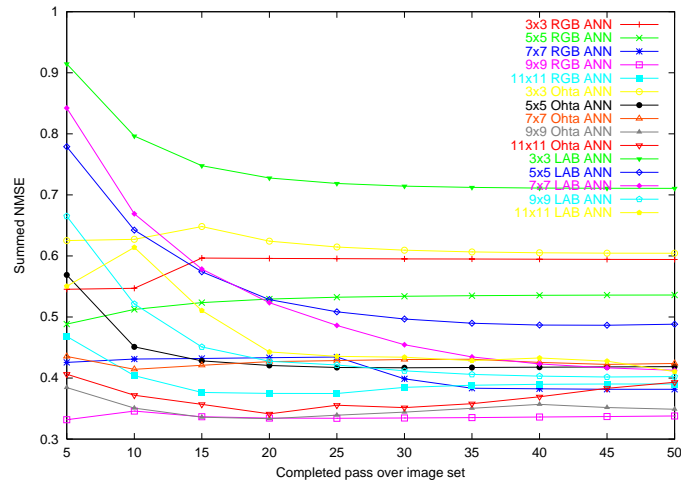


Figure C2.2d. Testing error for  $\sigma^2=1000$  Gaussian noise, plastic component image set.

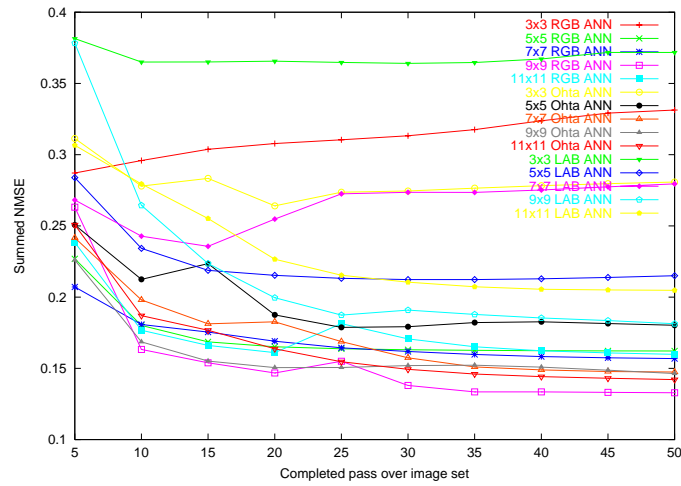


Figure C2.2e. Testing error for 2% and  $\sigma^2=100$  mixed noise, plastic component image set.

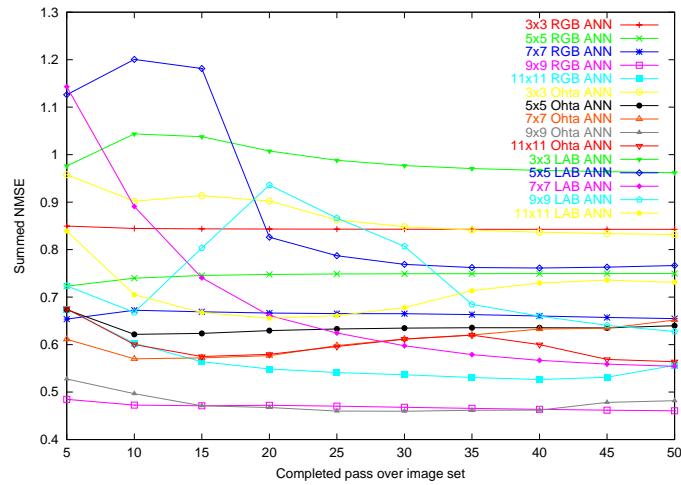


Figure C2.2f. Testing error for 5% and  $\sigma^2=1000$  mixed noise, plastic component image set.

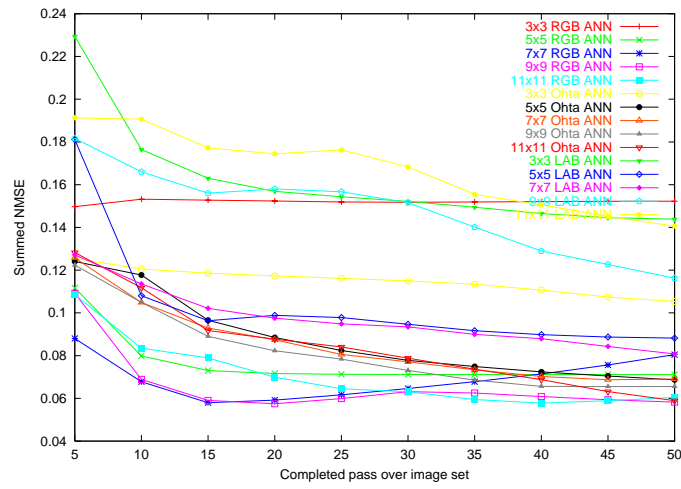


Figure C2.3a. Testing error for 2% impulse noise, confectionery image set.

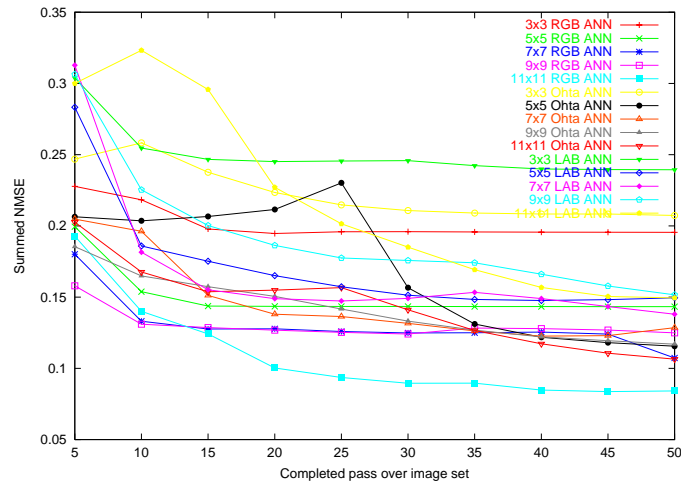


Figure C2.3b. Testing error for 5% impulse noise, confectionery image set.

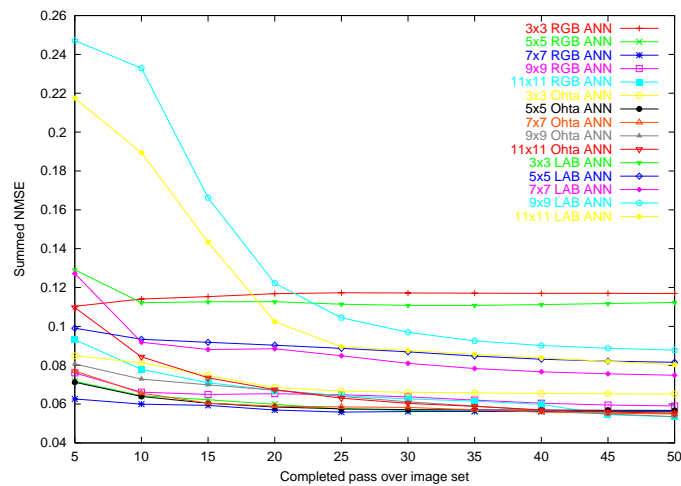


Figure C2.3c. Testing error for  $\sigma^2=100$  Gaussian noise, confectionery image set.

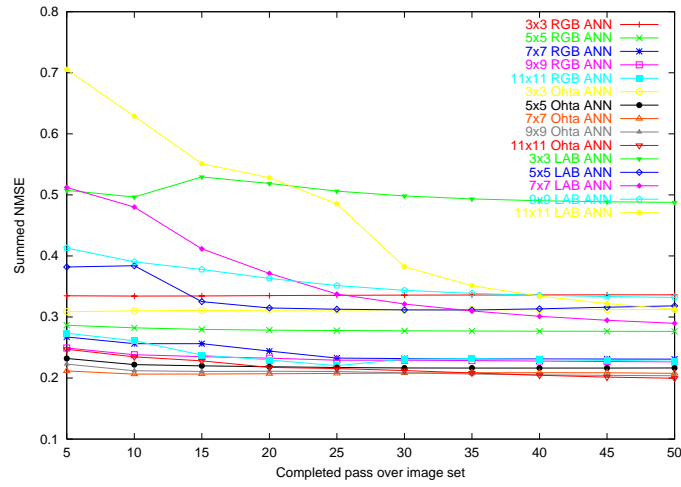


Figure C2.3d. Testing error for  $\sigma^2=1000$  Gaussian noise, confectionery image set.

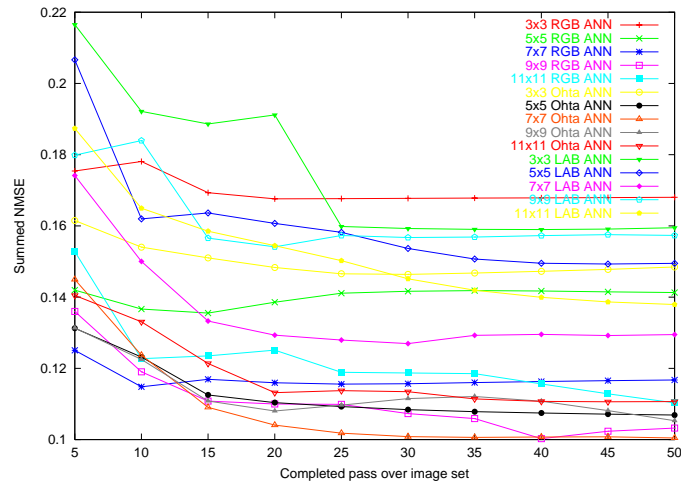


Figure C2.3e. Testing error for 2% and  $\sigma^2=100$  mixed noise, confectionery image set.

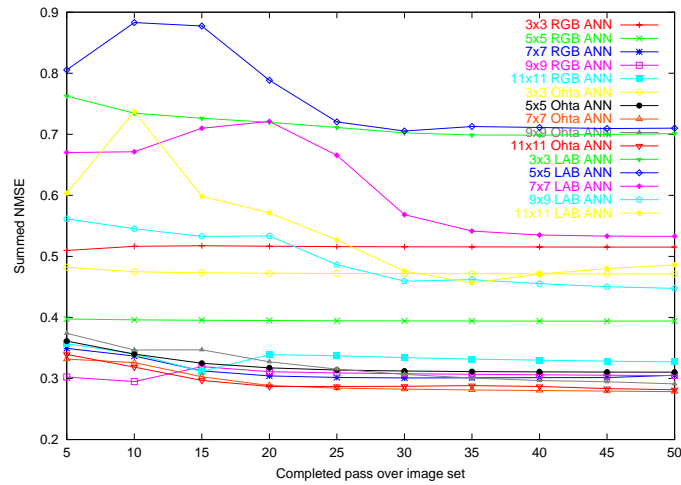


Figure C2.3f. Testing error for 5% and  $\sigma^2=1000$  mixed noise, confectionery image set.

### **C3 COMPARISON WITH CONVENTIONAL FILTERS**

To evaluate the efficacy of using a neural network to remove noise from colour images, the saved networks which performed the best are compared with an appropriate conventional filter, selected from the best performing of the standard VMF, AMF or  $\alpha$ VMF with circular operator sizes between  $3 \times 3$  and  $11 \times 11$ . The comparison is done on an image-by-image basis, and the results are shown overleaf in Figures C3.1a-f for the standard image set, Figures C3.2a-f for the plastic component image set and Figures C3.3a-f for the confectionery image set.

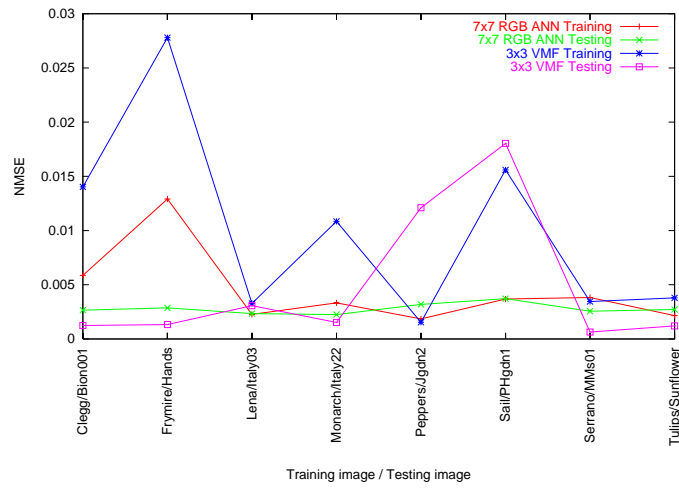


Figure C3.1a. Filter comparison for 2% impulse noise, standard image set.

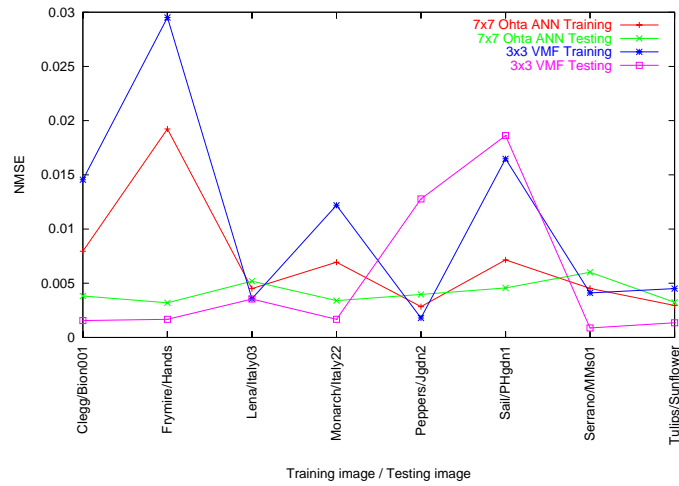


Figure C3.1b. Filter comparison for 5% impulse noise, standard image set.

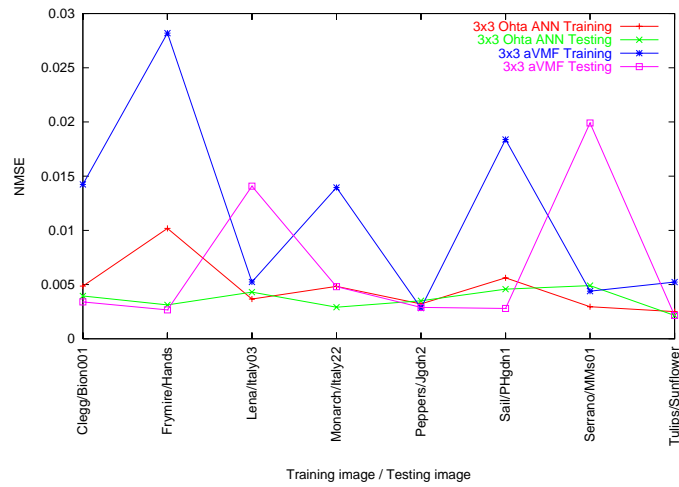


Figure C3.1c. Filter comparison for  $\sigma^2=100$  Gaussian noise, standard image set.



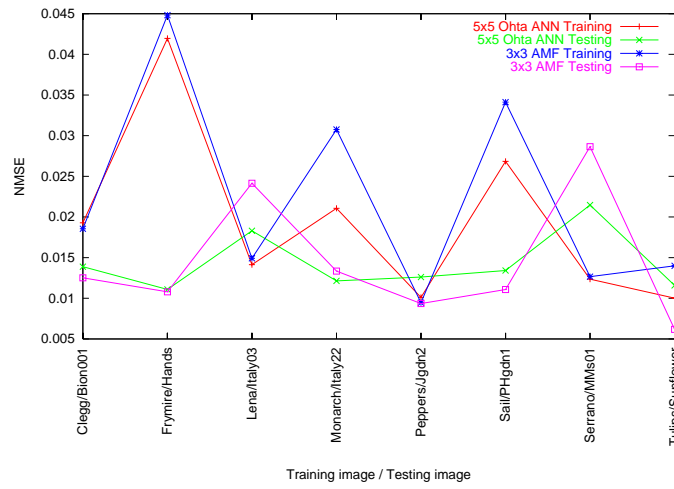


Figure C3.1d. Filter comparison for  $\sigma^2=1000$  Gaussian noise, standard image set.

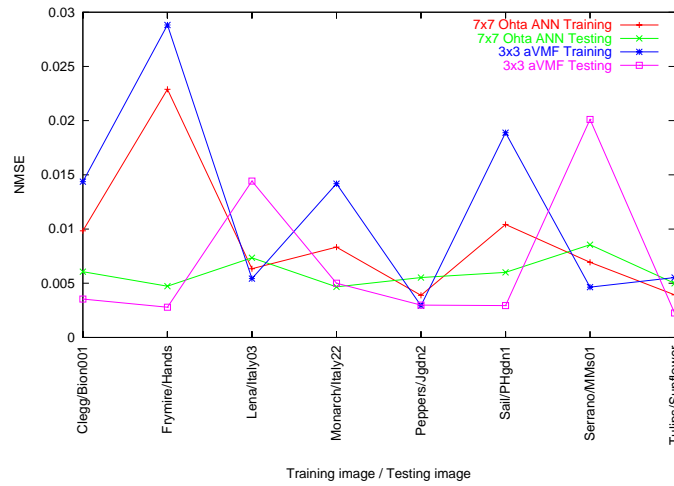


Figure C3.1e. Filter comparison for 2% and  $\sigma^2=100$  mixed noise, standard image set.

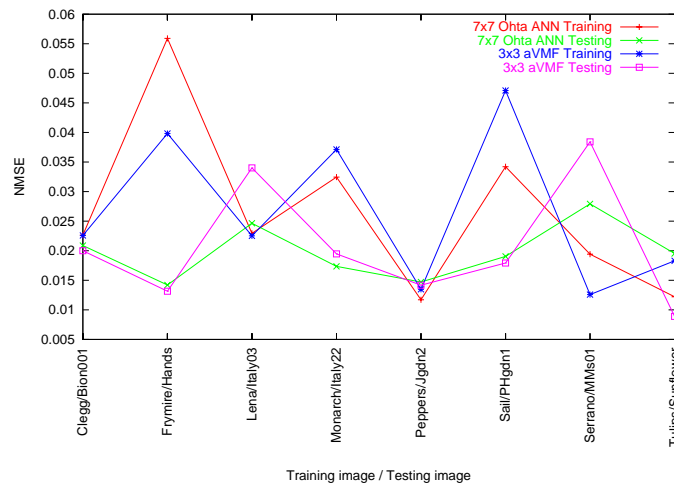


Figure C3.1f. Filter comparison for 5% and  $\sigma^2=1000$  mixed noise, standard image set.

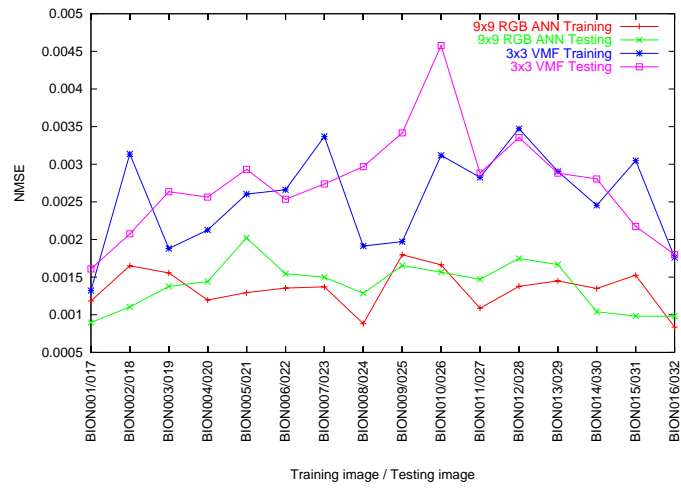


Figure C3.2a. Filter comparison for 2% impulse noise, plastic component image set.

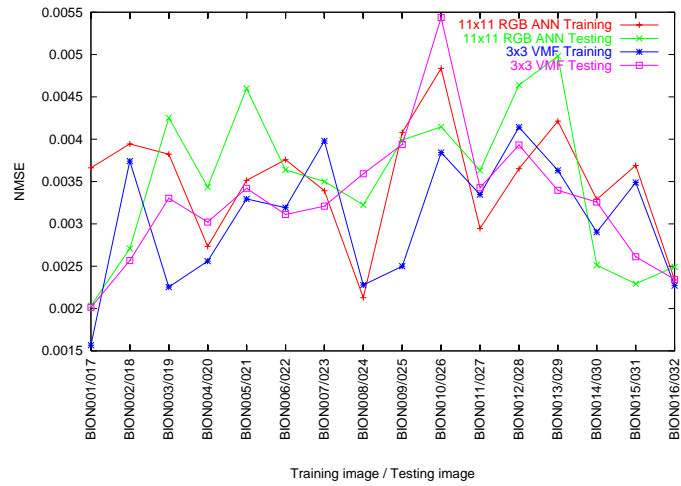


Figure C3.2b. Filter comparison for 5% impulse noise, plastic component image set.

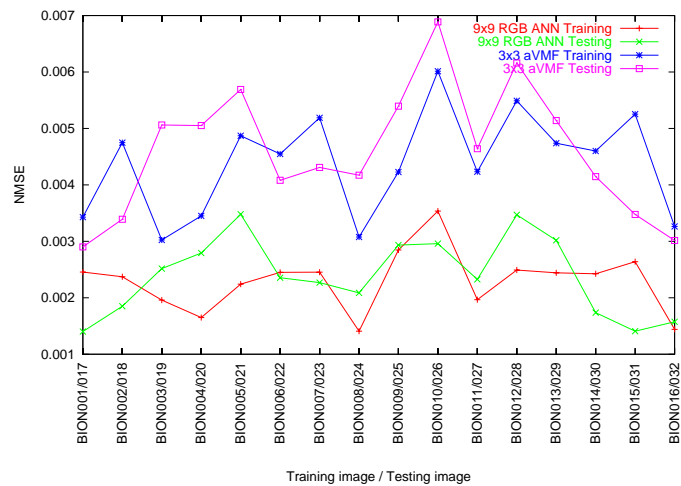


Figure C3.2c. Filter comparison for  $\sigma^2=100$  Gaussian noise, plastic component image set.

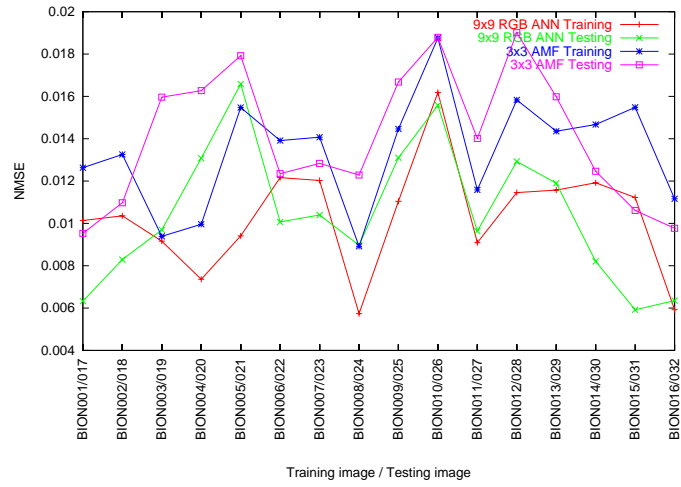


Figure C3.2d. Filter comparison for  $\sigma^2=1000$  Gaussian noise, plastic component image set.

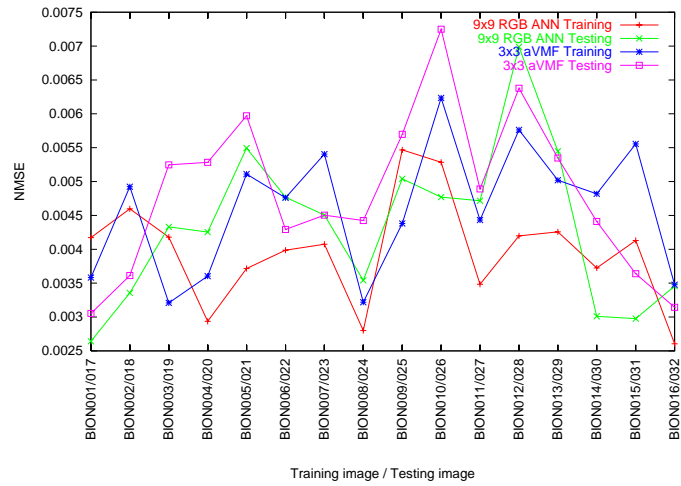


Figure C3.2e. Filter comparison for 2% and  $\sigma^2=100$  mixed noise, plastic component image set.

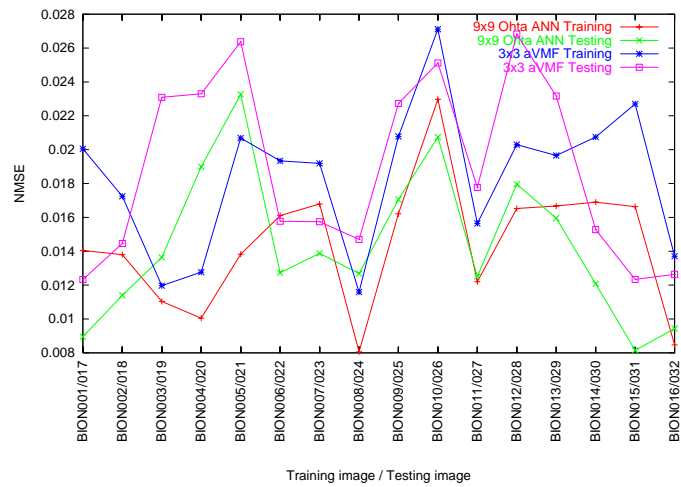


Figure C3.2f. Filter comparison for 5% and  $\sigma^2=1000$  mixed noise, plastic component image set.

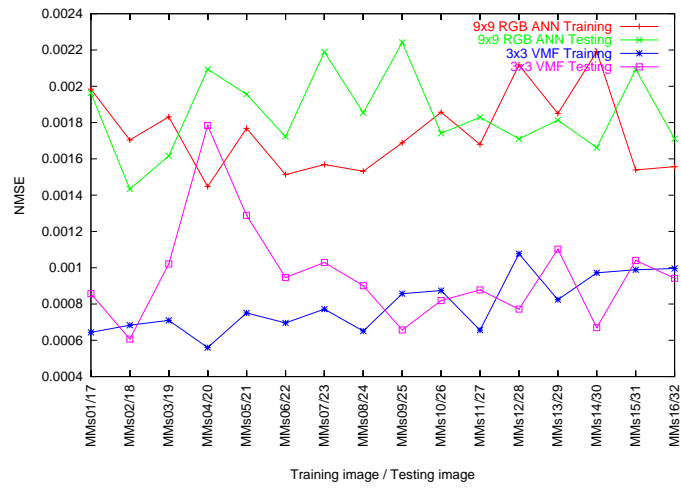


Figure C3.3a. Filter comparison for 2% impulse noise, confectionery image set.

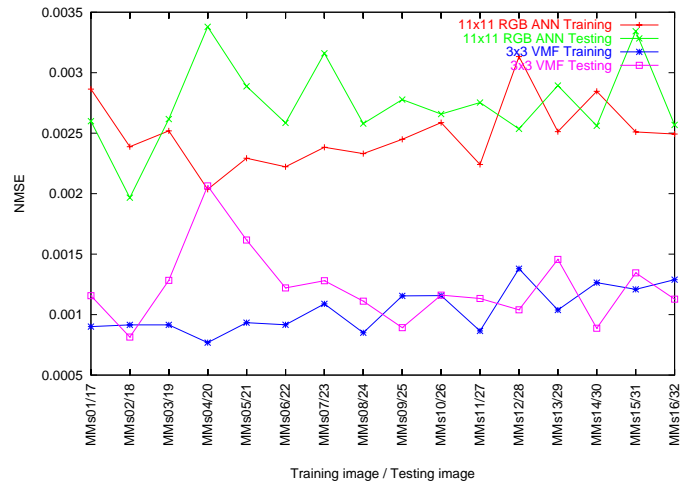


Figure C3.3b. Filter comparison for 5% impulse noise, confectionery image set.

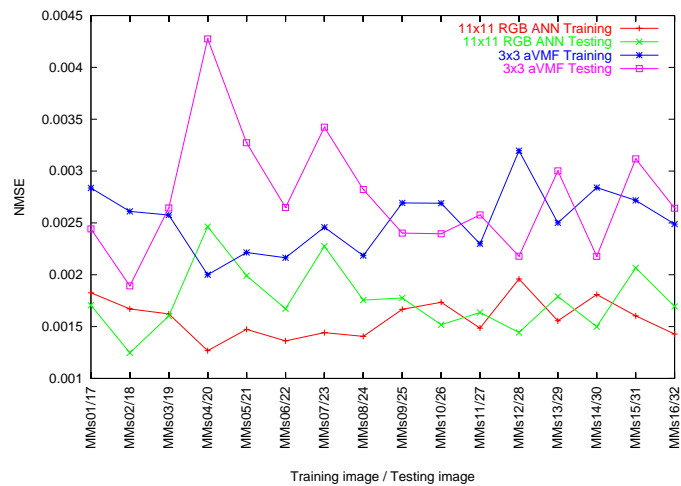


Figure C3.3c. Filter comparison for  $\sigma^2=100$  Gaussian noise, confectionery image set.

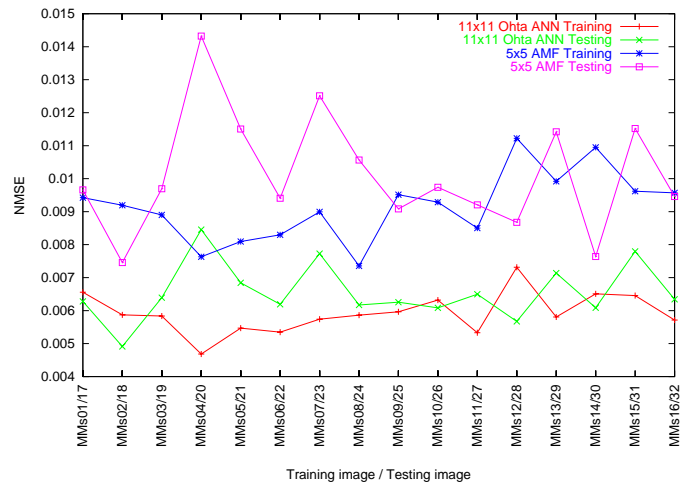


Figure C3.3d. Filter comparison for  $\sigma^2=1000$  Gaussian noise, confectionery image set.

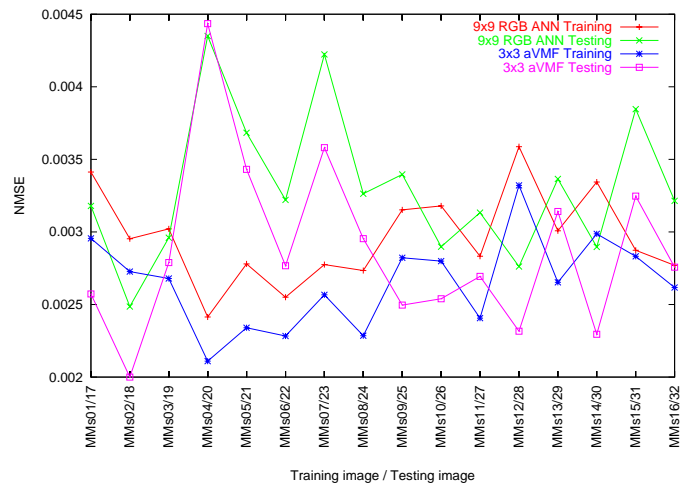


Figure C3.3e. Filter comparison for 2% and  $\sigma^2=100$  mixed noise, confectionery image set.

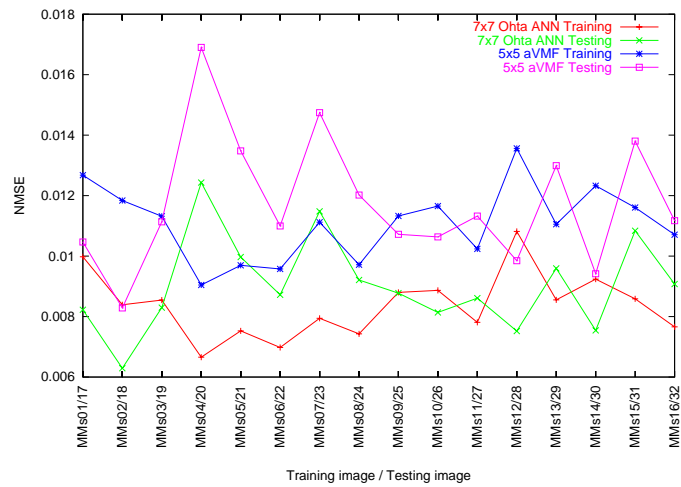


Figure C3.3f. Filter comparison for 5% and  $\sigma^2=1000$  mixed noise, confectionery image set.

# Appendix D

## INDEX OF TERMS

AMF	Arithmetic Mean Filter
ANN	Artificial Neural Network
ANNMF	Adaptive Nearest Neighbour Multichannel Filter
$\alpha$ VMF	Alpha Trimmed Vector Median Filter
BVDF	Basic Vector Directional Filter
CCD	Charge Coupled Device
CIE	Commission Internationale de l'Éclairage
CPU	Central Processing Unit
CRT	Cathode Ray Tube
DCT	Discrete Cosine Transform
DDF	Directional Distance Filter
DMVF	Directional Magnitude Vector Filter
DWT	Discrete Wavelet Transform
DWVMF	Distance Weighted Vector Median Filter
FFT	Fast Fourier Transform
GVDF	Generalized Vector Directional Filter
HMMF	Hybrid Mean-Median Filter

HSI	Hue Saturation Intensity
HSV	Hue Saturation Value
KLT	Karhunen-Loeve Transform
MCRE	Mean Chromaticity Error
MLP	Multi Layer Perceptron
(N)MSE	(Normalized) Mean Squared Error
(P)SNR	(Peak) Signal to Noise Ratio
RAM	Random Access Memory
RGB	Red Green Blue
SROF	Selective Rank Order Filter
T(V)MF	Truncated (Vector) Median Filter
VDF	Vector Directional Filter
VMF	Vector Median Filter

# Bibliography

1. Gonzalez, R. & Wintz, P. (1977). *Digital Image Processing*. Addison Wesley, Reading, MA.
2. Merbs, S. & Nathans, J. (1992) *Absorption spectra of human cone pigments*, Nature **356** pp433-5.
3. Jacobs, G. (1993) *The distribution and nature of colour vision among the mammals*, Biol. Rev. **68** pp413-71.
4. Webster, M. (1996) *Human colour perception and its adaptation*, Comput. Neural Syst. **7** pp587-634.
5. Polyak, S. (1957) *The Vertebrate Visual System*, University of Chicago Press, Chicago, IL.
6. Lennie, P., Pokorny J. & Smith V. (1993) *Luminance*, J. Opt. Soc. Am. **10** pp1283-93.
7. Kiver, M. (1965) *Color Television Fundamentals*, McGraw-Hill, New York.
8. Ramachandran, V. (1978) *Does colour provide an input to human motion perception?* Nature **275** pp55-6.
9. Livingstone, M. & Hubel D. (1987) *Psychophysical evidence for separate channels for the perception of the form, colour, movement and depth*, J. Neuroscience **7** pp3416-68.
10. Livingstone, M. & Hubel D. (1987) *Segregation of form, colour, movement and depth: Anatomy, physiology and perception*, Science **240** pp740-9.
11. Savoy, R.L., (1987) *Contingent after effects and isoluminance: Psychophysical evidence for separation of color, orientation, and motion*, Computer Vision, Graphics and Image Processing **37** pp3-19.



12. Troscianko, T. & Fahle, M. (1988) *Why do isoluminant stimuli appear slower?* J. Optical Soc. Am. pp871-80.
13. McIlhagga, W., Hine, T., Cole, G.R., & Snyder, A.W. (1990) *Texture segregation with luminance and chromatic contrast*, Vision Research **30**(3) pp489-95.
14. Sangwine, S.J. (1996) *Fourier transforms of colour images using quaternion or hypercomplex numbers*, Electronics Letters **32**(21) pp1979-80.
15. Panjwani, D. K., & Healey, G. (1995) *Markov Random Fields for unsupervised segmentation of textured color images*, IEEE Trans. Patt. Analysis and Machine Int. **17**(10) pp939-54.
16. Onyango, C.M., & Marchant, J.A. (1996) *Flexible colour point distribution models*, Image and Vision Computing **14** pp703-8.
17. Davies, E. R. (1997) *Machine Vision: Theory, Algorithms, Practicalities*, 2<sup>nd</sup> Ed., Academic Press, London.
18. Shufelt, J.A. (1997) *Texture analysis for enhanced color image quantization*, Graphical Models and Image Processing **59**(3) pp149-63.
19. Ahmad, I. S. & Reid, J.F. (1996) *Evaluation of colour representations for maize images*, J. agric. Engng Res. **63** pp185-96.
20. Song, K. Y., Kittler, J. & Petrou, M. (1996) *Defect detection in random colour textures*, Image & Vision Computing **14** pp667-83.
21. Saber, E., Tekalp, A. M., Eschbach, R. & Knox, K. (1996) *Automatic image annotation using adaptive color classification*, Graphical Models and Image Processing **58**(2) pp115-126.
22. Chang, C-C. & Wang, L-L. (1996) *Color texture segmentation for clothing in a computer-aided fashion design system*, Image & Vision Computing **14** pp685-702.

23. Zwiggelaar, R., Yang, Q., Garcia-Pardo, E. & Bull, C. R. (1996) *Use of spectral information and machine vision for bruise detection on peaches and apricots*, J. Agric. Engng Res. **63** pp323-32.
24. Slater, D. & Healey, G. (1997) *The illuminant-invariant matching of deterministic local structure in color images*, IEEE Trans. Patt. Analysis and Machine Int. **19**(10) pp1146-51.
25. Thakkar, S. & Huff T. (1999) *The Internet Streaming SIMD instructions*, Intel Technology Journal, Q2  
([http://www.intel.com/technology/itj/Q21999/PDF/simd\\_ext.pdf](http://www.intel.com/technology/itj/Q21999/PDF/simd_ext.pdf)).
26. Mittal, M. (1997) *MMX Technology Architecture Overview*, Intel Technology Journal, Q3  
(<http://www.developer.intel.com/technology/itj/q31997.htm>).
27. AMD, (March 2000) *3Dnow! Technology Manual*, Document no. 21928G/0,  
([http://www.amd.com/us-en/assets/content\\_type/white\\_papers\\_and\\_tech\\_docs/21928.pdf](http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/21928.pdf))
28. Sangwine, S.J. & Horne, R.E.N. (eds) (1998) *The Colour Image Processing Handbook*, Chapman & Hall, London.
29. Berriss, H. (2000) *Acquisition of Skin Wound Images and Measurement of Wound Healing Rate and Status using Colour Image Processing*, PhD thesis, University of Reading.
30. Li, C.H. & Yuen, P.C. (2002) *Tongue image matching using color content*, Pat. Rec. **35** pp407-419.
31. Derganc, J., Likar B. & Bernard R., et al. (in press) *Real-Time Automated Visual Inspection of Color Tablets in Pharmaceutical Blisters*.

32. ISO/IEC (1994) *ISO/IEC Standard 10918-1:1994 Information Technology – Digital compression and coding of continuous-tone still images: requirements and guidelines*, Geneva, (<http://www.iso.ch/>, <http://www.iec.ch/>)
33. Shannon, C.E. (1948) *A Mathematical Theory of Communication*, Bell System Technical Journal **27** pp379-423, 623-656.
34. Strømme, Ø. (1999) *On the Applicability of Wavelet Transforms to Image and Video Compression*, PhD thesis, University of Strathclyde.
35. Skodras, A.N., Christopoulos, C.A. & Ebrahimi, T. (2001) *JPEG2000: The upcoming still image compression standard*, Pat. Rec. Let. **22** pp1337-1345.
36. Hsu, T., Calway, A.D. & Wilson, R. (1993) *Texture Analysis using the Multiresolution Fourier Transform*, 8<sup>th</sup> Scandinavian Conference on Image Analysis, Tromsø, Norway.
37. Oppenheim, A.V., & Schafer, R.W. (1999) *Discrete-Time Signal Processing*, Prentice Hall, New Jersey.
38. Ohta, Y. (1985). *Knowledge-based Interpretation of Outdoor Natural Color Scenes*. Pitman Publishing, London, England.
39. Gonzalez, R.C. & Woods, R.E. (1992). *Digital Image Processing*. Addison Wesley, Reading, MA.
40. Travis, D. (1991) *Effective Colour Displays: Theory and Practice*. Academic Press.
41. Cheng, H.D., Jiang, X.H. & Wang, J. (2002) *Color image segmentation based on homogram thresholding and region merging*, Pat. Rec. **35** pp373-393.
42. Scheering, C. & Knoll, A. (1997) *Fast colour image segmentation using a pre-clustered chromaticity-plane*, ICASSP '97, pp3145-3147.
43. Pitas, I. & Kiniklis, P. (1996) *Multichannel Techniques in Color Image Enhancement and Modeling*, IEEE Trans. Im. Proc. **5**(1) pp168-171.

44. Tan, T.S.C. & Kittler, J. (1993) *Colour texture classification using features from colour histogram*, 8<sup>th</sup> Scandinavian Conference on Image Analysis, Tromsø, Norway, pp807-813.
45. Sobottka, K. & Pitas, I. (1996) *Extraction of facial regions and features using color and shape information*, Proc. Int. Conf. Im. Proc. **3** pp 483-486.
46. Lee, C.H., Kim, J.S. & Park, K.H. (1996) *Automatic human face location in a complex background using motion and color information*, Pat. Rec. **29** pp1877-1889.
47. Foley, J.D., van Dam, A., Feiner, S.K. & Hughes, J.F. (1997) *Computer Graphics: Principles and Practice*, 2<sup>nd</sup> Edition, Addison Wesley.
48. Pascho, G. & Valavanis, K.P. (1999) *A color texture-based visual monitoring system for automated surveillance*, IEEE Trans. Syst. Man. Cybernet. **29**(1), pp298-307.
49. ITU (1994) *ITU-R Recommendation BT.601-5: Studio Encoding Parameters of Digital Television for standard 4:3 and wide-screen 16:9 aspect ratios*, Geneva, (<http://www.itu.ch/>)
50. ITU (1990) *ITU-R Recommendation BT.709-2: Parameter Values for the HDTV Standards for production and International Programme Exchange*, Geneva, (<http://www.itu.ch/>)
51. CIE colorimetry (1986) *Official Recommendations of the CIE*, 2<sup>nd</sup> Edition, Pub. No. 15.2, Vienna, Austria.
52. Wysecki, G. & Stiles, W.S. (1982) *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2<sup>nd</sup> Edition, John Wiley, New York.
53. Pitas, I. & Kiniklis, P. (1992) *Multichannel Histogram Equalization with Applications in the HSI Color Space*, Sig. Proc. VI: Theories and Applications, Elsevier Science, pp1469-1472.
54. Heckbert, P. (1982) *Color image quantization for frame buffer display*, Computer Graphics, **16**(3), pp297-304.

55. Yang, C.C. & Rodriguez, J.J. (1997) *Efficient luminance and saturation processing techniques for color images*, Journal of Visual Communication and Image Representation, **8**(3) pp263-77.
56. Schettini, R. (1992) *Low-level Segmentation of Complex Color Images*, Sig. Proc. VI: Theories and Applications, Elsevier Science, pp535-538.
57. Schumeyer, R., Heredia, E. & Barner, K. (1997) *Region of interest priority coding for sign language video conferencing*, Proc. IEEE 1<sup>st</sup> Workshop on Multimedia Sig. Proc., pp531-536.
58. Cheng, H.D., Jiang, X.H. & Wang, J. (2001) *Color image segmentation: advances and prospects*, Pat. Rec. **34** pp2259-2281.
59. Drimbarean, A. & Whelan, P.F. (2001) *Experiments in colour texture analysis*, Pat. Rec. Lett. **22** pp1161-1167.
60. Verikas, A., Malmqvist, K. & Bergman L. (1997) *Colour image segmentation by modular neural network*, Pat. Rec. Lett. **18** pp173-185.
61. Swenson, R.L. & Dimond, K.R. (1998) *A universal colour transformation architecture*, Pat. Rec. Lett. **19** pp805-813.
62. Zaccarin, A. & Bede, L. (1993) *A Novel Approach for Coding Color Quantized Images*, IEEE Trans. Im. Proc. **2**(4) pp442-453.
63. Kato, Z., Pong, T-C. & Lee J.C-M. (2001) *Color image segmentation and parameter estimation in a markovian framework*, Pat. Rec. Lett. **22** pp309-321.
64. McQueen, J.B. (1967) *Some methods of classification and analysis of multivariate observations*, Proc. 5<sup>th</sup> Berkeley Symp. On Math. Stat. Prob. pp181-197
65. Sapiro, G. (1997) *Color Snakes*, Comp. Vis. And Im. Understanding **68**(2) pp247-253.
66. Shafarenko, L., Petrou, M., & Kittler, J. (1998) *Histogram-Based Segmentation in Perceptually Uniform Color Space*, IEEE Trans. Im. Proc. **7**(9) pp1354-1358

67. Gervautz, M. & Purgathofer, W. (1990) *A Simple Method for Color Quantization: Octree Quantization*, in Glassner, A.S. (Ed.) *Graphics Gems*, Academic Press, San Diego, CA pp287-293.
68. Dekker, A. (1994) *Kohonen neural networks for optimal colour quantization*, Network: Comp. in Neural Syst. **5** pp351-367.
69. Davies, E.R. (1989) *Edge location shifts produced by median filters: theoretical bounds and experimental results*, Signal Process. **16**(2), pp83-96.
70. Glasbey, C.A and Jones, R. (1997) *Fast computation of moving average and related filters in octagonal windows*, Pat. Rec. Lett. **18** pp555-565.
71. Nvidia Corp. (2000) *Anisotropic Filtering*  
[http://www.nvidia.com/docs/IO/1324/ATT/filter\\_anisotropic2.doc](http://www.nvidia.com/docs/IO/1324/ATT/filter_anisotropic2.doc)
72. Kang, K., Astola, J. & Neuvo, Y. (1995) *Nonlinear Multivariate Image Filtering Techniques*, IEEE Trans. Im. Proc. **4**(6), pp788-797.
73. Chehdi, K. (1992) *An Approach To The Automatic Filtering of Images*, Sig. Proc. VI: Theories and Applications, Elsevier Science, pp1409-1412.
74. Astola, J., Haavisto, P. & Neuvo, Y. (1990) *Vector Median Filters*, Proc. IEEE **78**(4), pp678-689.
75. Tukey, J.W. (1977) *Exploratory Data Analysis*, Addison-Wesley, Menlo Park, CA.
76. Barnett, V. (1976) *The Ordering of Multivariate Data*, J. Statist. Soc. Amer. A, **139**(3) pp318-354.
77. Vardavoulia, M.I., Andreadis, I. & Tsalides, P. (2001) *A new vector median filter for colour image processing*, Pat. Rec. Lett. **22** pp675-689.
78. Fotopoulos, S., Sindoukas, D., Laskaris, N. & Economou, G. (1997) *Image enhancement using color and spatial information*, ICASSP '97 pp2581-2584.
79. Trahanias, P.E. & Venetsanopoulos, A.N. (1993) *Color Edge Detection Using Vector Order Statistics*, IEEE Trans. Im. Proc. **2**(2) pp259-264.

80. Lau, D.L & Gonzalez, J.G. (1997) *The closest-to-mean filter: an edge preserving smoother for Gaussian environments*, ICASSP '97 pp2593-2596.
81. Barni, M., Cappellini, V. & Mecocci, A. (1992) *The use of different metrics in vector median filtering: application to fine arts and paintings*, Sig. Proc. VI: Theories and Applications, Elsevier Science, pp. 1485-1488.
82. Plataniotis, K.N., Androutsos, D. & Venetsanopoulos, A.N. (1997) *Content-based colour image filters*, Elec. Lett. **33**(3) pp202-205.
83. Sebe, N. & Lew, M.S. (2001) *Color-based Retrieval*, Patt. Rec. Lett. **22** pp223-230.
84. Restrepo, A. & Bovik, A.C. (1988) *Adaptive trimmed mean filters for image restoration*, IEEE Trans. Acoustics Speech and Sig. Proc. **36**(8) pp1326-1337.
85. Bednar, J.B. & Watt, T.L. (1984) *Alpha-trimmed means and their relationship to median filters*, IEEE Trans. Acoustics Speech and Sig. Proc. **32**(1) pp145-153.
86. Trahanias, P.E. & Venetsanopoulos, A.N. (1993) *Vector Directional Filters – A New Class of Multichannel Image Processing Filters*, IEEE Trans. Im. Proc. **2**(4) pp528-534.
87. Trahanias, P.E., Karakos, D. & Venetsanopoulos, A.N. (1996) *Directional Processing of Color Images: Theory and Experimental Results*, IEEE Trans. Im. Proc. **5**(6) pp868-880.
88. Karakos, D.G. & Trahanias, P.E. (1995) *Combining vector median and vector directional filters: The directional-distance filters*, Proc. IEEE Conf. Image Proc., ICIP '95 pp171-174.
89. Ekman, G. (1963) *A direct method for multidimensional ratio scaling*, Psychometrika **28** pp33-41.
90. Plataniotis, K.N., Androutsos, D. & Venetsanopoulos, A.N. (1996) *Fuzzy adaptive filters for multichannel image processing*, Sig. Proc. **55**(1) pp93-106

91. Bardos, A.J. & Sangwine, S.J. (1998) *Measuring noise in colour images*, Digest of IEE Colloquium on Non-linear Signal & Image Processing, 22<sup>nd</sup> May 1998.
92. McCulloch, W.S. & Pitts, W. (1943) *A logical calculus of the ideas immanent in nervous activity*, Bulletin of Mathematical Biophysics **5**, pp115-133.
93. Rosenblatt, F. (1958) *The perceptron: a probabilistic model for information storage and organisation in the brain*, Psychol. Review **65**, pp386-408
94. Rosenblatt, F. (1962), *Principles of Neurodynamics*, Spartan, New York.
95. Rumelhart, D.E. & McClelland, J.L. (1986) *Parallel Distributed Processing*, MIT Press, MA.
96. Srinivasan V., Bhatia P. & Ong S.H. (1994) *Edge detection using a neural network*, Pat. Rec. **27**(12) p1653.
97. Molina, C., Bastard, D., Baylou, P. & Najim, M. (1992) *New approach for multi-perceptron architecture design applied to the edge detection problem*, Sig. Proc. VI: Theories and Applications, Elsevier Science, pp1219-1222.
98. Campbell, N.W., Thomas, B.T. & Troscianko, T. (1997) *Automatic Segmentation and Classification of Outdoor Images using Neural Networks*, Int. J. Neur. Syst. **8**(1) pp137-144.
99. Campbell, N.W., Thomas, B.T. & Troscianko, T. (1996) *Segmentation of Natural Images using Self-Organizing Feature Maps*, British Machine Vision Conference, Sep 1996, BMVA pp223-232.
100. Campbell, N.W. & Thomas, B.T. (1996) *Neural Networks for the Segmentation of Outdoor Images*, Int. Conf. Eng. App. Neur. Net., Systems Engineering Association, June 1996 pp343-346.
101. Setchell, C.J. & Campbell, N.W. (1999) *Using Colour Gabor Texture Features for Scene Understanding*, 7<sup>th</sup> Int. Conf. Im. Proc. App. IEE, pp372-376.



102. Maillard, E., Zerr, B. & Merckle, J. (1992) *Classification of texture by an association between a perceptron and a self-organizing feature map*, Sig. Proc. VI: Theories and Applications, Elsevier Science, pp1173-1176.
103. Annadurai, S. & Balasubramaniam, A. (1996) *Supervised feedforward fuzzy artificial neural network for handwritten alphanumeric character recognition*, Elec. Lett. **32**(21) pp1987-1989.
104. Wood, M.E.J., Campbell, N.W. & Thomas, B.T. (1997) *Searching Large Image Databases using Radial Basis Function Neural Networks*, Proc. 6<sup>th</sup> Im. Proc. App. IEE, pp116-120.
105. Zoroofi, R.A., Taketani, H., Tamura, S., Sato, Y. & Sekiya, K. (2001) *Automatic inspection of IC wafer contamination*, Pat. Rec. **34** pp1307-1317.
106. Xin, Y., Boekaerts, P., Bister, M., Taeymans, Y. & Cornelis J. (1992) *Left ventricular boundary detection in echocardiographical images using artificial neural networks*, Sig. Proc. VI: Theories and Applications, Elsevier Science, pp. 1199-1202.
107. Kotropoulos, C., Augé, E. & Pitas, I. (1992) *Two-layer learning vector quantizer for color image quantization*, Sig. Proc. VI: Theories and Applications, Elsevier Science, pp1177-1180.
108. DeNatale, F.G.B., Desoli, G.S., Fioravanti, S. & Giusto, D.D. (1992) *Neural prediction in image vector quantization*, Sig. Proc. VI: Theories and Applications, Elsevier Science, pp1215-1218
109. Figueiredo, M.A.T. & Leitão, J.M.N. (1994) *Sequential and Parallel Image Restoration: Neural Network Implementations*, IEEE. Trans. Im. Proc. **3**(6) pp789-801.
110. Greenhill, D. & Davies, E.R. (1994) *Relative effectiveness of neural networks for image noise suppression*, Pat. Rec. Prac. IV, pp367-378.
111. Hopfield, J.J. (1982) *Neural networks and physical systems with emergent collective computational abilities*, Proc. Natl. Acad. Sci. **79** pp2554-2558

112. Kohonen, T. (1989) *Self-Organization and Associative Memory*, Springer-Verlag, New York.
113. Grossberg, S.A. & Carpenter, G.A. (1987) *ART2: self-organization of stable category recognition codes for analog input patterns*, *Applied Optics* **26**(23) pp4919-4930
114. Grossberg, S.A. (1988) *Neural Networks and Natural Intelligence*, MIT Press, Cambridge, MA.
115. Brownrigg, D.R.K. (1984) *The weighted median filter*, *Commun. Assoc. Comput. Mach.* **27**, pp807-818
116. Sun, T. & Neuvo, Y. (1992) *A simple synthesis method for weighted median filters*, *Sig. Proc. VI: Theories and Applications*, pp1405-1408
117. Song, X., Yin, L. & Neuvo, Y. (1992) *Image sequence coding using adaptive weighted median prediction*, *Sig. Proc. VI: Theories and Applications*, pp1307–1310
118. Plataniotis, K.N., Androutsos, D. & Venetsanopoulos, A.N. (1995) *Colour image processing using fuzzy vector directional filters*, *Proc. IEEE Workshop on Nonlinear Signal/Image Processing*, Neos Marmas, Greece, pp535-538
119. Davies, E.R. (1999) *Image distortions produced by mean, median and mode filters*, *IEE Proc. Vision Image Signal Process.* **146**(5), pp279-285
120. Tang, K., Astola, J. & Neuvo, Y. (1992) *Multivariate ordering and filtering with application to color image processing*, *Sig. Proc. VI: Theories and Applications*, pp1481-1484
121. Reinhard, E., Shirley, P. & Troscianko, T. (2001) *Natural image statistics for computer graphics*, Technical report UUCS-01-002, School of Computing, Univ. of Utah, Salt Lake City.
122. Simoncelli, E.P. & Olshausen, B.A. (2001) *Natural Image Statistics and Neural Representation*, *Ann. Review of Neuroscience* **24**, pp1193-1216.

123. Barni, M. (1997) *A Fast Algorithm for 1-Norm Vector Median Filtering*, IEEE Trans. Im. Proc. **6**(10), pp1452-1455.
124. Barni, M. (1994) *Fast vector median filter based on Euclidean norm approximation*, IEEE Sig. Proc. Lett. **1**, pp92-94.
125. Pitas, I. (1993). *Digital Image Processing Algorithms*, Prentice Hall International Series in Acoustics, Speech and Signal Processing, Prentice Hall, New York.
126. McConnell, S. (1993) *Code Complete*, Microsoft Press, VA pp679-82
127. Ziv, J. & Lempel, A. (1977) *A universal algorithm for sequential data compression*, IEEE Trans. Info. Theory **23**(3), pp337-343
128. Ziv, J. & Lempel, A. (1978) *Compression of individual sequences via variable-rate coding*, IEEE Trans. Inf. Theory **24**(5), pp53-60
129. Welch, T.A. (1984) *A technique for high performance data compression*, IEEE Computer **17**(6), pp8-19
130. Yang, G.J. & Huang, T.S. (1981) *The effect of median filtering on edge location estimation*, Comput. Graph. Image Process. **15**, pp224-245
131. Bovik, A.C., Huang, T.S. & Munson, D.C. (1987) *The effect of median filtering on edge estimation and detection*, IEEE Trans. Pattern Anal. Mach. Intell. **9**(2), pp. 181-194
132. Coleman, G.B. & Andrews, H.C. (1979) *Image segmentation by clustering*, Proc. IEEE **67**, pp. 773-785
133. Evans, A.N. & Nixon, M.S. (1995) *Mode filtering to reduce ultrasound speckle for feature extraction*, IEE Proc. Vision Image Signal Process., **142**(2), pp. 87-94
134. Davies, E.R. (1988) *On the noise suppression and image enhancement characteristics of the median, truncated median and mode filters*, Patt. Rec. Lett. **7**(2), pp. 87-97

135. Davies, E.R. (1992) *Simple two-stage method for the accurate location of Hough transform peaks*, IEE Proc. E **139**(3), pp. 242-248
136. Davies, E.R. (1997) *Shifts produced by mode filters on curved intensity contours*, Electronics Lett. **33**(5), pp. 381-382
137. Davies, E.R. (1991) *Median and mean filters produce similar shifts on curved boundaries*, Electronics Lett. **27**(10), pp. 826-828
138. Nieminen, A., Heinonen, P. & Neuvo, Y. (1987) *A new class of detail-preserving filters for image processing*, IEEE Trans. Pattern Anal. Mach. Intell. **9**(1), pp. 74-90
139. Fukunaga, K. & Hostleiter, L.D. (1975) *The estimation of the gradient of a density function, with applications in pattern recognition*, IEEE Trans. Info. Theory **21**(1), pp32-40
140. Cheng, Y. (1995) *Mean shift mode seeking and clustering*, IEEE Trans. Pattern Anal. Mach. Intell. **17**, pp. 790-799
141. Comaniciu, D. & Meer, P. (1999) *Distribution Free Decomposition of Multivariate Data*, Pattern Analysis and Applications **2**, pp22-30
142. Greenhill, D.R. (1994) *Investigation of image segmentation problems using training techniques*, PhD Thesis, Royal Holloway University of London.
143. Nightingale, C. & Hutchinson, R.A. (1990) *Artificial neural nets and their application to image processing*, Brit. Telecom Tech. J. **8**(3), pp81-93
144. Weber, M., Crilly, P.B. & Blass, W.E. (1991) *Adaptive noise filtering using an error-backpropagation neural network*, IEEE Trans. Instrumentation and Measurement **40**, pp820-825
145. Klimasauskas, C. (1989) *Neural nets and noise filtering*, Dr. Dobb's Journal **14**, pp32-48.
146. Pham, D.T. & Bayro-Corrochano, E.J. (1992) *Neural computing for noise filtering, edge detection and signature extraction*, Journal of Systems Engineering **1**, pp13-23.

147. Cybenko, G. (1989) *Approximation by superpositions of a sigmoidal function*, Mathematics of Control, Signals and Systems **2**(4), pp303-314
148. Eberhart, R.C. & Dobbins, R.W. (eds) (1990) *Neural Network PC Tools: A Practical Guide*, Academic Press, San Diego, CA.
149. Bateman, M. (1996) *Application of Neural Networks to Image Processing*, MSc Thesis, Royal Holloway University of London.
150. Ciftcioglu, Ö. & Türkcan, E. (1992) *Selection of hidden layer nodes in neural networks by statistical tests*, Sig. Proc. VI: Theories and Applications, Elsevier Science, pp1181-1184.
151. University of Waterloo, *Waterloo Repertoire Colorset*,  
<http://links.uwaterloo.ca/colorset.base.html>, Montreal, Canada.
152. Petzold, C. (1999) *Programming Windows*, 5<sup>th</sup> Ed., Microsoft Press, WA.
153. Eng, H-L. & Ma, K-K. (2001) *Noise Adaptive Soft-Switching Median Filter*, IEEE Trans. Im. Proc. **10**(2), pp242-251.