Investigating Polynomial Fitting Schemes for Image Compression

By

Salah Ameer

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2009

Author's Declaration Page

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Image compression is a means to perform transmission or storage of visual data in the most economical way. Though many algorithms have been reported, research is still needed to cope with the continuous demand for more efficient transmission or storage. This research work explores and implements polynomial fitting techniques as means to perform block-based lossy image compression.

In an attempt to investigate nonpolynomial models, a region-based scheme is implemented to fit the whole image using bell-shaped functions. The idea is simply to view an image as a 3D geographical map consisting of hills and valleys. However, the scheme suffers from high computational demands and inferiority to many available image compression schemes. Hence, only polynomial models get further considerations.

A first order polynomial (plane) model is designed to work in a multiplication- and division-free (MDF) environment. The intensity values of each image block are fitted to a plane and the parameters are then quantized and coded. Blocking artefacts, a common drawback of block-based image compression techniques, are reduced using an MDF line-fitting scheme at blocks' boundaries. It is shown that a compression ratio of 62:1 at 28.8dB is attainable for the standard image PEPPER, outperforming JPEG, both objectively and subjectively for this part of the rate-distortion characteristics.

Inter-block prediction can substantially improve the compression performance of the plane model to reach a compression ratio of 112:1 at 27.9dB. This improvement, however, slightly increases computational complexity and reduces pipelining capability. Although JPEG2000 is not a block-based scheme, it is encouraging that the proposed prediction scheme performs better in comparison to JPEG 2000, computationally and qualitatively. However, more experiments are needed to have a more concrete comparison.

To reduce blocking artefacts, a new postprocessing scheme, based on Weber’s law, is employed. It is reported that images postprocessed using this scheme are subjectively more pleasing with a marginal increase in PSNR (<0.3 dB). The Weber’s law is modified to perform edge detection and quality assessment tasks.

These results motivate the exploration of higher order polynomials, using three parameters to maintain comparable compression performance. To investigate the impact of higher order polynomials, through an approximate asymptotic behaviour, a novel linear mapping scheme is designed. Though computationally demanding, the performances of higher order polynomial approximation schemes are comparable to that of the plane model. This clearly demonstrates the powerful approximation capability of the plane model. As such, the proposed linear mapping scheme constitutes a new approach in image modeling, and hence worth future consideration.

Acknowledgment

I would like to express my deep gratitude to my supervisor Dr. Otman Basir for his support, guidance, and patience.

I am also thankful to the examination committee for their valuable feedback and to the external examiner for agreeing to read this dissertation.

I would like to thank PAMI students and friends for their participation in the two surveys shown in the results of this work.

The support of OGS for five consecutive terms is also acknowledged.

I feel I should express my appreciation to the encouragement and support of my wife and parents. In particular, I'm very thankful to the patience of my wife, during this long period and limited financial resources.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The increasing demands for high volume data transfer and/or storage necessitate the need for a mechanism to perform such operations in a more compact form. This mechanism is called compression where appropriate data model is selected for implementation together with some coding scheme. As an example, the transmission of a coloured video signal composed of 30 frames/sec may require a bandwidth of 188.7 Mbit/sec for a frame size of 512x512 pixels. This amount is highly beyond the standard 6Mbit/sec allocated to commercial channels.

Compression is some times termed as redundancy reduction. Hence, many data redundancies should be exploited to attain the highest possible compression rate. Redundancies can be between adjacent pixels or within a certain region of the image whether it is regularly shaped like blocks or irregular to represent actual objects (or subobjects) in the image. Redundancies can be due to some geometrical relations between the positions or gray values of some pixels. A more important form of redundancies is that corresponding to psychovisual aspects of the human visual system (HVS) where some degradations cannot be perceived by the eye.

In general, compression falls in two categories lossless and lossy. The former produces an exact replica of the original data and may be termed as exact reconstruction or reversible transformation) [Berg and Mikhael 1994]. The latter, on the contrary, introduces errors and the reconstructed data is not the original any more.

In both compression schemes, the procedure consists of three major steps as illustrated in Fig 1.1. These three steps are typically implemented in the following order: decorrelation, quantization, and coding. In the decorrelation step, a certain transformation is performed to produce uncorrelated coefficients. The coefficients in the transform domain are then quantized, ranging from simple thresholding to the optimum or Max quantizer, to reduce the number of allocated bits. In fact, the error is mainly due to this step and hence, it bears the distinction between lossy and lossless compression. In fact, almost negligible errors can be produced due to transformation roundoff errors. After performing quantization, the resulting quantized coefficients are of different probabilities and an efficient coding scheme (step 3) can further reduce the number of required bits.

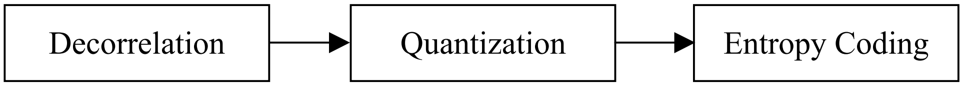


Fig 1.1 Data compression steps

In some applications, like computer file transfer and medical image archiving, errors are prohibited (lossless compression) and hence low (around 2:1) compression ratios (CR), size of original data to that of the reconstructed data, are common in practice. In others, speech and image lossy compression for example, some error margin is tolerated on the expense of quality degradation at the decoder. Fortunately, sometimes not, human visual system is nonlinear and eyes can tolerate certain errors. Hence, for image compression applications, high compression ratio can be obtained with perceptually no difference between the original and reconstructed images.

Revisiting Fig 1.1, decorrelation in an image or a video is typically performed using spatial and/or temporal pixel prediction or block, typically of 8x8 pixels, transformation (typically linear). Traditionally, prediction can be performed between adjacent pixels in the same frame and/or between adjacent pixels in consecutive video frames to produce uncorrelated prediction errors. On the other hand, block-based coding transforms the intensities and/or chromatic information of each block, usually nonoverlapped with its neighbours, into uncorrelated coefficients in the transform domain. A more compact form, though more complex, is to consider the whole image as consisting of regions (overlapped or not) representing the various objects (or subobjects) in the scene. Many features, e.g. colour, edginess, texture and motion, can be incorporated for further improvements.

In all lossy compression schemes, performance is judged according to an objective quality called peak signal to noise ratio PSNR, given CR is the same between the competing schemes. Unfortunately, PSNR correlation with human visual system is widely criticized throughout the image processing community. This criticism is due to two facts. First, difference in PSNR does not necessarily means a comparable difference in quality. Second, in some situations, a PSNR value can give preference to a certain image that is the worst due to perceptual comparison. Hence, some subjective quality measure, depending on some viewers, may be performed to support the PSNR value.

Many compression schemes are available in the literature; the next chapter tries to describe some of these algorithms. However, theoretically speaking, region-based compression schemes have high potential [Sikora, 2005] in exceeding the current barrier on the value of compression ratio. Unfortunately, current implementations of region-based schemes are computationally intensive. In a somewhat contrary characteristic, block-based schemes are typically fast but are saturated at least from compression-wise performance. Pixel-based or predictive coding cannot compete as a stand-alone scheme but rather as a supporting tool.

As mentioned in the previous paragraph, block-based schemes are saturated. To jump over this barrier, polynomial fitting schemes are investigated on a regularly shaped region, a fixed size block, in the least-square sense. The author is not aware of any previous image compression work on plane or polynomial fitting for fixed size blocks. There are, however, some work on implementing plane models for variable sized blocks, quadtrees to be more specific. To reduce computational cost and to allow future merging with more complex algorithms, like region-based for example, the ultimate objective is to design a multiplication- and division-free implementation. The multiplication-free characteristic should prevail in the postprocessing stages also.

Two computationally efficient plane (first-order surfaces) models are proposed, implemented, and compared to existing algorithms. Typical block size is 8x8 pixels. Multiplication-free feature is maintained in both schemes. The first model (basic scheme) is implemented on nonoverlapping blocks of the image. A plane, $\boldsymbol{a}x + \boldsymbol{b}y + \boldsymbol{c}$, is independently describing each block. The performance is superior (objectively and subjectively) to that of JPEG at a compression ratio of ~60:1. In addition, the computational cost is significantly lower.

To further improve the compression performance, with tolerated degradation in quality, the second (prediction scheme) is proposed. It uses the basic scheme formulation with a constraint to evaluate two parameters of each block, except for the top-left one, are dependent on a pixel value from the boundary of a neighbouring block. The two parameters are ***c*** and ***a***(***b***) depending on the location of the neighbouring block. The previous discussion on the support of predictive coding is used here. In fact, the proposed prediction is better than using that of JPEG where ***c*** is predicted from the ***c***’s of neighbouring blocks. It should be emphasized that the proposed form of prediction has not been reported previously. Despite the low computational cost, the prediction scheme performed better than JPEG2000 both objectively and subjectively. In both plane models, quantization is performed using empirically optimized parameters. Huffman coding is used. However, to further increase compression ratio, coding is performed on the combined symbol (***a*** , ***b***).

Higher order polynomial fitting schemes (no prediction) are also investigated. The performance, restricting to the three-parameter case, is at most comparable to that of the plane. In fact, an asymptotic behaviour is approximated, using a linear mapping scheme, and implemented showing negligible improvements over the basic plane fitting model. This clearly shows the power of the plane model despite its simplicity. Nevertheless, the proposed linear mapping scheme is a new form of describing or modelling image blocks.

In addition, new post-processing algorithms are presented with some details due to their high influence on the reconstruction quality. Of special importance is the postprocessing due to the incorporation of a proposed edge detection scheme using Weber fraction. The proposed edge detection scheme has a better impact on the subjective quality compared to Sobel operator for example. Interestingly, Weber law can be modified further to act as a quality assessment scheme.

The organization of this work is as follows; chapter 2 presents a literature survey on the general subject of image compression. Chapter 3 illustrates some work on using nonpolynomial functions to describe images. The proposed plane models are described in chapters 4 and 6. Chapter 5 describes some higher order models and schemes that can be used independently or as supplementary tools to those described in chapters 4 and 6. Postprocessing, using modified Weber law, is presented in chapter 7. In addition chapter 7 contains some results on applying Weber law in edge detection, and quality assessment. Results and comparisons are illustrated after the theoretical description of each scheme. Finally, conclusions and future work are presented in chapter 8.

Chapter 2

Background and Literature Survey

Image compression has been widely investigated and many algorithms have been proposed [Sikora 2005] and [Egger *et al* 1999]. Comparison between algorithms is often based on two aspects: compression ratio and reconstruction quality. Compression ratio CR is defined as

$$CR = \frac{No.\ of\ bits\ in\ original\ file}{No.\ of\ bits\ in\ compressed\ file} \tag{2.1}$$

Quality, can also be measured subjectively based on some scoring procedure (obtained by employing a group of viewers) or objectively as Peak Signal to Noise Ratio (PSNR):

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2\mathrm{XY}}{\sum_x\sum_y\left(\mathrm{g(x, y)} - \mathrm{\hat{g}(x, y)}\right)^2}\right) \tag{2.2}$$

where g and $\hat{g}$ are the original and reconstructed images respectively, $x = 1, \ldots X$, $y = 1, \ldots Y$, and X and Y are image dimensions.

The essential difference between image compression algorithms is the implementation of the decorrelation step, see Fig 1.1; hence it is more convenient to consider the other two steps first, namely, quantization and coding.

2.1 Quantization

In reference to Fig 1.1, the output of the decorrelation step, could be a scalar or a vector, should be quantized. The desired compression ratio can be attained through an appropriate selection of quantization steps. In other words, the output can only belong to a predefined set. These steps can be uniform or not, depending on the statistical characteristics of the quantity to be quantized. If the probability density function (PDF), *p(x)*, of the parameter under consideration is known, the optimum least square (Lloyd-Max) quantizer can be described by [Max 1960]:

$$t_k = \frac{r_k + r_{k-1}}{2} \quad \text{and} \quad r_k = \int_{t_k}^{t_{k+1}} xp(x) \Bigg/ \int_{t_k}^{t_{k+1}} p(x) \tag{2.3}$$

where r_k is the reconstruction level of the interval (t_k , t_{k+1}) and t_k is a threshold. Nevertheless, when the PDF follows a uniform distribution, the above quantizer reduces to the linear or uniform quantization given by:

$$\begin{aligned} t_k &= t_1 + (k - 1)q \\ r_k &\equiv t_k + q/2 \\ q &= \left(t_{L+1} - t_1\right)/L \end{aligned} \tag{2.4}$$

where L is the number of quantization levels.

2.2 Encoding

The easiest way to encode a set of symbols is to assume that they occur with equal repetitions, resulting in an equal code length. A more efficient way is to use variable length coding, i.e., the code length is inversely proportional to the frequency of occurrence of the encoded symbol.

The simplest variable-length coding scheme is the comma code. The most frequent symbol gets the code 0(1), the next most frequent 10(01), and so on until the least frequent symbol, which gets 1…10(0…01). A more compact and efficient form is through entropy coding namely: Huffman or arithmetic coding [Shi and Sun 2000]. In the former, a coding table is built starting from the least frequent symbol upwards till all symbols get their codes. High probability symbols get short codes while low probability ones get longer codes. In arithmetic coding, however, a fixed code is sent that represents a variable number of symbols. Each code combines the symbols in proportion to their probabilities. Header bits (in entropy coding) are required to define the coding table or the symbol probabilities to the decoder. This requirement can be relaxed if symbols' probabilities are slightly fluctuating with time and a fixed encoding table can be used.

Going back to Fig 1.1, the decorrelation step can be considered as the distinguishing step between different image compression schemes. The description will be specific to most of the algorithms in image compression literature as presented in the following sections.

2.3 Image Compression Schemes

This section gives a general description of the main image compression schemes available in the literature. The schemes can be categorized into four subgroups according to the processing element. These subgroups are: pixel-based, block-based, subband-based, and region-based. The following subsections present these subgroups.

2.3.1 Predictive Coding

In differential pulse code modulation (DPCM) [Habibi 1977], a pixel is predicted from a subset (a, b and c) of its causal neighbourhood (see Fig 2.1) and prediction error is quantized and coded. The prediction is simply a weighted sum of all elements in a selected neighbourhood.

```
. . . . . . . . . . . . .
. . . . . c b d . . . . .
. . . . . a x . . . . . .
. . . . . . . . . . . . .
```

Fig 2.1 Causal neighbourhood of pixel x

Predictive coding is preferred for its simplicity in video coding since consecutive frames are highly correlated. It was found [Habibi 1971] that improvements are marginal for a correlation distance of more than 3 pixels. In practice (lossless JPEG compression), one of the 7 predictors, listed in Table 2.1, or a combination of them is used. High compression (more than 8:1) is difficult to attain due to accumulated errors and the need for multi-model prediction.

Table 2.1 Predictors used in lossless JPEG scheme (see Fig 2.1).

N	Predictor $\hat{x}$
1	a
2	b
3	c
4	a + b – c
5	a + (b – c)/2
6	b + (a – c)/2
7	(a + b)/2

Although lossless image compression is beyond the scope of this work, there are rather advanced prediction schemes to perform such a task, see [Zhang and Adjeroh, 2008], [Solé and Salembier, 2007], and [Kingston and Autrusseau, 2008].

2.3.2 Block-Based Coding

To overcome the limitations of predictive coding, block based (dividing the image into nonoverlapping blocks) compression techniques were suggested [Jain 1981]. However, at higher compression rates, these techniques suffer from visually annoying artefacts at block boundaries and some post-processing is therefore needed.

Block based techniques can be categorized into training type and non-training type techniques. Training type techniques include: vector quantization VQ [Li and

Zhang 1995], and neural networks NN [Jiang 1999] and [Kim and Lee 2002]. Iterated functions or fractals [Wohlberg and DeJager 1999] can be considered as a category of VQ with a virtual codebook composed of blocks surrounding the current block. Non-training type techniques include: block truncation coding BTC [Delp and Mitchell 1979], transform coding TC (e.g., Discrete Cosine Transform DCT [Furht 1995]), and surface fitting [Eden *et al* 1986].

2.3.2.1 Training Type Techniques

In these techniques, some offline learning or training is required before applying these techniques. In VQ and NN, compression performance is dependent on how similar is the image to the training set.

The general idea behind VQ is to use some set of images (or blocks of one image) as representatives for other images preferably of some sort of similarity with the training set. These images are divided into smaller blocks, typically of size 4x4 or 8x8 pixels, to be used in building the codebook through training or clustering process. More complex schemes, like the genetic algorithm GA, can be used to design an optimum codebook. The codebook contains few vectors (blocks) that should be available for both the encoder and decoder. Compression is achieved by sending few bits (per block) representing the index of the best matching block to the current one. The matching process is computationally demanding even when nonexhaustive search is performed. Many training methods were implemented to enhance the representation of the codebook to the data set [Feng *et al.* 2007] and [Tsekouras *et al.* 2008]. The decoding time is much less than that of the encoding, since no comparison is needed [Laha *et al.* 2008].

In a somewhat similar fashion to VQ, NN use these building blocks to train a multilayer neural network. After completing the training phase and convergence is reached, the final NN structure is divided such that input-hidden weights are at the encoder side, while the decoder contains hidden-output weights. The standard way in performing compression [Jiang 1999] is by using a smaller number of hidden nodes. In my opinion, however, a better way to achieve compression is by quantizing (typically uniform) the outputs of the hidden neurons. The output of each node is a nonlinear function of the weighted sum of outputs from the previous layer. Another form is through unsupervised clustering using self-organizing Kohonen model [Soliman and Omari, 2006], [Abbas 2007], and [Dokur 2008]. Other forms can be obtained by modifying the node type in a layer, e.g., a multiplicative instead of an additive one, as in [Iyoda *et al.* 2007].

Fractal compression, on the other hand, uses the image's self-similarity to construct a "virtual codebook". Each block (called range block) has its codebook defined by some surrounding larger domain blocks that may be overlapped and/or not. Iterative affine (scaling, rotation, shifting and down sampling) transformations are applied on any initial conditions, other image or even randomly generated pixel values, to obtain an approximate version of the original image. The information is hence captured by the affine parameters that are quantized to reach the desired compression. The computational burden is often high, especially at the encoding stage. Several attempts were made to improve the speed of the fractal image compression scheme [Distasi *et al.* 2006], [Jackson *et al.* 2007], [Zhou *et al.* 2008], and [Tseng *et al.* 2008].

2.3.2.2 Non-Training Type Techniques

One of the simplest approaches is the block truncation coding BTC [Dasarathy 1995]. Each block, typically of size 4x4 pixels, is considered to come from some random distribution of pixel values. The idea is to divide the block pixels into two groups each has its own reconstruction level. The first and second moments, or the two reconstruction levels, are sent together with the block map (1 bit per pixel indicating its group). The mathematical derivation is simply to preserve the first 3 moments of a vector X resulting in [Delp and Mitchell 1979]

$$\begin{aligned} a &= E\{X\} - \sigma\sqrt{p/(m-p)} \\ b &= E\{X\} + \sigma\sqrt{(m-p)/p} \\ p &= m\left(1 + A\sqrt{1/\left(1 + A^2\right)}\right)/2 \end{aligned} \tag{2.5}$$

where $\sigma^3 A = 3\mathrm{E}\{\mathrm{X}\}\mathrm{E}\{\mathrm{X}^2\} - 2\mathrm{E}\{\mathrm{X}^3\}\mathrm{E}\{\mathrm{X}\}^3$, σ is the standard deviation, *m* is the total number of points, *p* is a threshold (number of pixels having level *a*), and *a* and *b* are the reconstruction levels. Unfortunately, CR cannot go beyond 8:1 without hybridizing with other schemes like VQ [Mohamed and Fahmy 1995], fuzzy edge detection [Amarunnishad *et al.* 2008], and pattern fitting [Dhara and Chanda 2004] and [Dhara and Chanda 2007].

In transform coding, on the other hand, a linear transformation is used to map the current block into the transform domain producing uncorrelated coefficients. The energy is concentrated in few transform coefficients, typically the lowest in frequency, which are quantized and efficiently coded to perform compression. The transformation can be generally described by

$$\begin{aligned} F &= Tf \\ f &= T^{-1}F = T^{t}F \end{aligned} \tag{2.6}$$

where T is the transformation matrix and f and F are the original and transformed blocks respectively. T is a unitary transformation and hence, the transpose is the inverse. The previous equation governs all transform coding techniques; the special case of discrete cosine transform DCT is given (for 8x8 blocks) by

$$F(u,v) = \frac{1}{4}\left(\frac{1}{\sqrt{2}}\right)^{\delta(u)+\delta(v)} \sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos\left(\frac{(2x+1)u\pi}{16}\right)\cos\left(\frac{(2y+1)v\pi}{16}\right) \tag{2.7}$$

where $\delta(n)$ is the Dirac delta or impulse function. The transformed coefficients $F(u, v)$ are quantized, depending on the quantization table used, and sent in a zigzag order. In fact, the performance of any block-based transformation is upper-bounded by the optimum or KL transform. However, due to the computational complexity of KL transform and its data dependency, DCT is preferred [Jain 1981], [Alkholidia *et al.* 2007], and [Ponomarenko *et al.* 2007]. There is a wealth of literature on applying transform coding in image compression, like Hartley transform [Sunder *et al.* 2006], fuzzy transforms [Di Martino and Sessa 2007], Łukasiewicz transform [Di Nola and Russo 2007], and 3D matrix transform [Zhang *et al.* 2008].

At high compression rates, however, transform coding techniques suffer from blocking artefacts at blocks' boundaries and a post-processing step is necessary at the decoder to enhance the reconstruction quality.

2.3.3 Wavelet (Subband) Compression

Subband coding (wavelets) [Lin and Vaidyanathan 1996] differs from block-based techniques in performing the transformation on the whole image rather than part of it. However, some techniques operate on large blocks. Hence, it has less blocking artefacts; however, the reconstructed image tends to be blurry. Nevertheless, its performance is much better than traditional block-based techniques [Kaur *et al.* 2006] and [Bruni and Vitulano 2007].

The subbands are constructed through successive filtering-downsampling (upsampling at the decoder) [Lin and Smith 2008]. This technique can be viewed as performing block processing in the frequency domain. In wavelet image compression [Shapiro 1993], the image is decomposed into four bands, namely: LL, LH, HL, and HH. L and H correspond to a low and a high pass filter respectively. The high pass filter can be obtained by subtracting the low pass filtered output from the original image. The sequence of filter application is arbitrary, i.e., we can apply filtering to the horizontal and then to the vertical direction and vice versa. This is why we describe the filter using two letters. A downsampling by a factor of two then follows. The process can be applied for several times on the resulting LL (LL_1 to be more accurate) band to obtain LL_2, LH_2, HL_2, HH_2 and so on. The coefficients at each band are then quantized to reach the desired compression. Since the energy is concentrated in the LL band, larger quantization steps are used in other bands and hence a small portion of the bit budget is allocated to high frequency subbands [Chang and Girod 2007], [Yang *et al.* 2008].

Many filter models have been proposed in the literature [Lin and Vaidyanathan 1996] for wavelet image compression. The JPEG2000 standard uses the Daubechies (9,7) filter due to its superior empirical performance on a wide range of images. To reduce computational burden, the filter is broken to four 2x2 matrix multiplications using the lifting scheme [Acharya and Tsai 2005].

2.3.4 Region-Based Compression

Traditional transform coding techniques are saturated as far as compression ratio is concerned [Gilge 1990] and [Kaup and Aach 1994]. Region- or segmentation-based techniques were suggested to exceed this barrier [Kaup and Aach 1998] and to support new multimedia services [Salembier and Marqués 1999]. The term “second generation” is often used to indicate their superior performance over the previously mentioned schemes [Cermelli *et al* 1994], [Salembier and Pardas 1994] and [Sikora 2005]. It has been shown [Biggar *et al* 1988] and [Ran and Farvardin 1995] that at higher rates, reconstruction quality of region-based techniques exceeds that of DCT.

The general idea of region- or segmentation-based compression is to divide the image into regions that are not necessarily of regular shape. These regions are commonly constructed using some clustering or segmentation procedure that depends on the pixel gray (colour) value and/or motion [Salembier and Marqués 1999] parameters. Each region is represented by two codes. The first (preferably a chain code) describes the location of boundary pixels. The second represents the best approximation of the region enclosed by this boundary. In addition, coding gains can be obtained by avoiding repetition of the common boundary points between adjacent regions. The usual compromise between quality and CR is dependent to a certain extent on the number of regions.

Watershed algorithm, morphological operators [Bosworth and Acton 2000], [Salembier *et al* 1996] and [Salembier and Pardas 1994], region growing [Yemez *et al* 2005], genetic algorithms GA [Aravind *et al* 2002] combined with gradient information, and many other algorithms [Muñoz *et al* 2003] can be used to segment the image into different regions. A preprocessing can be used to better describe the texture regions [Hussain and Reed 1994] through some statistical test to separate edges from uniform or texture regions.

After segmentation, each region can now be approximated in one of many different ways. [Biggar *et al* 1988], for example, implemented up to second order surfaces to describe each region. A similar procedure is to use some basis functions [Gilge 1990] followed by an orthogonalization routine. Coloured regions [Roterman and Porat 2003] can be described according to HVS sensitivity by describing G component using a 2nd order polynomial, while R and B components are constructed using a linear function of G. Successive approximation (from an orthonormal set) was implemented in [Kaup and Aach 1994] independently on each region. A linear system is then solved to find the weights corresponding to the selected orthogonal functions. In [Salembier *et al* 1996] a suggestion was made to implement successive approximation between frames and/or resolution layers. VQ can also be combined with polynomials [DeNatale *et al* 1995] to reach a better compromise between quality and performance. [Kaup and Aach 1998] implemented a DCT scheme defined on the smallest MxN circumscribing rectangle.

2.4 Surface Fitting

This technique can be fused in some of the techniques described in the previous section. In the general field of image processing, surface (or polynomial) fitting has been used in image segmentation [Lim and Park 1988], image noise reduction [Sinha and Schunck 1992] and quality improvement of block-based compression [Kieu and Nguyen 2001] and [Laha *et al* 2004]. Lost subband coefficients [Hemami and Gray 1997] can be reconstructed by fitting the known samples to some surface. Splines can be used [Baseri and Modestino 1994] to encode the lowest frequency band in subband coding. RBF networks [Kim and Lee 2002] can be combined with surface fitting to perform compression using a predefined set of patterns for the centres. The term surface fitting was also used by [Chen *et al* 1994] to describe successive mean approximation.

Polynomial fitting was implemented [Cabrelli and Molter 1990] in contour coding of black and white images. Splines were used in block-based compression [Watanabe 1997] to preserve continuity between the pixels inside the block. Image representation by verge (high curvature) points [Wang *et al* 2005] is an elegant suggestion to emphasize the importance of boundary pixels (edges) in producing perceptually pleasant pictures. Zigzag scan was used in [Nguyen and Oommen 1997] to convert the block into

1D. Following that, where moment preserving is used as the objective criterion, piecewise linear fitting is performed between knots. Another implementation of polynomial fitting is in predicting motion compensation vectors in video coding [Karczewics *et al* 1997].

Segmentation-based [Biswas 2003] image compression also uses 1D and 2D polynomial fitting. The former is used to encode boundary points while the latter to approximate slowly varying areas enclosed by these points. To reduce complexity, slowly varying regions are usually approximated by a constant intensity depending on the split and merge technique. A flexible way of constructing variable size triangular blocks through split and merge was implemented in [Lu *et al* 2000] and [Demaret *et al* 2006].

2.4.1 Compression Via Surface Fitting (Quadtree)

In [Eden *et al* 1986], a mathematical framework for using polynomial interpolation in image processing was presented. To achieve compression, least square approximation was presented as an optimization algorithm when the number of coefficients is required to be less than the number of pixels in a block. Mathematically, we have

$$Minimize \ \sum_{x}\sum_{y}\left(z(x,y)-g(x,y)\right)^{2} \qquad (2.8)$$

where *g(x, y)* is the original intensity (or any colour component) value and *z(x, y)* is the suggested (polynomial) function. However, only separable cases were considered in

[Eden *et al* 1986]. Also, computations could be further reduced if the origin of the coordinate system is the block centre as will be emphasized later.

A simplified derivation for first order (plane) fitting was proposed by [Strobach 1991] in the form

$$\underset{a,b,c}{Minimize} \sum_{x}\sum_{y}\left(ax+by+c-g(x,y)\right)^{2} \tag{2.9}$$

The coefficients ***a***, ***b*** and ***c*** of a 2Nx2N block are computed from their NxN counter parts and are assumed uniformly distributed. The resulting values of the above minimization procedure, on the largest possible blocks usually 32x32 or 64x64, are retained if the resulting error is less than a predescribed threshold. Otherwise, the block is split into four blocks and the minimization process is repeated again. A PSNR of 32 dB was reported for 16:1 compression (0.5 bits per pixel bpp) with high complexity in building the quadtree describing sizes of the compressed blocks. To reduce the error energy imposed by quantizing ***a*** and ***b***, the block centre was selected as the origin of the coordinate system. In fact, the selection of the origin can also affect the range of ***c*** as will be shown later in the experiments.

The computation of 2Nx2N parameters from their NxN counter parts can be generalized for higher order polynomials [Philips 1991]. A related quadtree approach was proposed by [Hasegawa and Yamasaki 2002] to predict block corners from the upper left one. These four corners are used in the decoder to find the coefficients of (***d***xy + ***a***x + ***b***y + ***c***).

Strobach technique was independently generalized to video compression by using a zero order term for the time axis [Wellimen *et al* 1991]. Another attempt, called incomplete polynomial transform, is proposed by [Aydinoglu and Hayes 1996] to quantize the coefficients, resulting from fitting a 4x4 block, through subband coding. A first order model was suggested by [Lan *et al* 1998] to compress videos; however, the parameters were uniformly quantized.

2.5 Video Compression

Video is simply a sequence of images. Therefore, all the previously described schemes can be implemented on each frame (motion JPEG [Westwater and Furht 1997]) or generalized to 3D (3D JPEG [Westwater and Furht 1997]). A more common approach, called motion compensation, is to encode the displacement vectors describing the movements of objects from frame to frame. Vector components need not be integers, requiring interpolation (usually linear), and ½ or ¼ pixels are common in practice for motion prediction.

Block matching [Jain and Jain 1981] is widely accepted due to its reduced computational burden and increased coding efficiency. In this approach, the current frame is partitioned into nonoverlapping blocks (usually of size 16x16 and to less extent 8x8 pixels) that are matched (within some predefined neighbourhood) to blocks in the reference frame, typically the previous one. Full or exhaustive search, in the matching process, has some benefits (global optimum can be found), however, other searches like logarithmic search [Bhaskaran and Konstantinides 1997] may be preferred due to its reduced computational complexity. An efficient search strategy in the form of quadtrees (multiresolution or gaussian pyramid) is usually preferred. After finding an acceptable match and assuming uniform motion throughout the block, prediction error and translation (difference in centre coordinates) parameters are encoded (usually after a DCT routine).

Another approach in performing motion compensation is through pixel recursive methods [Musmann *et al* 1985]. Displacement vectors are found on pixel basis through nonlinear optimization. The goal is to minimize the prediction error given by

$$\sum_{x,y\in\Omega} W(x,y)\big(g(x-d_x,y-d_y,n-1)-g(x,y,n)\big)^2 \tag{2.10}$$

where $\underline{\boldsymbol{d}} = (d_x, d_y)$ is the displacement vector, $g(x, y, n)$ is the intensity value at pixel (x, y) in frame n, $W(x, y)$ is some weighting function, and Ω is a predefined neighbourhood (could be as small as 1 pixel).

2.6 Objective versus Subjective Image Quality Assessment

Quality assessment (QA) in image processing has been an active area of research in the past two decades. Several survey and comparative papers have been published on QA schemes, e.g., [Eskicioglu and Fisher, 1993], [Eskicioglu and Fisher, 1995], [Eudu and Mayache, 1998], [Al-Otum, 2003], [Eckert and Bradley, 1998], [Winkler, 1999], and [Meesters *et al*, 2004]. In fact, [Avcibas *et al*, 2002] and [Sheikh *et al*, 2006] introduce a statistical measure to assess QA schemes.

The subjective quality of a compressed image is a viewer-centred criterion that is tedious, time consuming and mathematically intractable. In fact, as pointed by Ridder, some of the procedures implemented to find the mean opinion score (MOS) are biased [Ridder, 1996]. The former can be further subdivided in accordance with the value of correlation the scheme has with the human visual system (HVS).

It is widely accepted that the traditional mean square error (MSE) and peak signal to noise ratio (PSNR) are of poor correlation with the eye perception in the sense that a difference in the measure do not necessarily contribute to a similar difference in quality. What can make the situation worse is that the two schemes (MSE and PSNR) can produce a value favouring an image that is more annoying to the observer perception. Hence, a quest for a measure having a better correlation with the human eye is certainly beneficial. One of the early ideas is to compensate for the discrepancy between the conventional MSE and the HVS by considering a weighted difference between original and distorted images [Marmolin, 1986] and [Fuhrmann *et al*, 1995]. A general formula can be stated as

$$Index = \frac{1}{N}\left(\sum_{i=1}^{N} w(i)\left|g(i) - \hat{g}(i)\right|^{\beta}\right)^{1/\beta} \tag{2.11}$$

where N is the number of pixels, w is a weighing function, and g and $\hat{g}$ are the original and distorted images respectively.

Due to the complexity of HVS, Equation (2.11) can give good performance and sometimes outperforms other complex schemes for certain image types, as demonstrated by many researchers, e.g., [Choy *et al*, 1996], [Elbadawy *et al*, 1998], and [Zampolo and Seara, 2005]. Interestingly, despite the inferiority (on average) to other schemes, a scheme described by (2.11) can easily lend itself to a rate distortion optimization in the compression domain.

In a similar fashion, a weighted difference can be calculated in the frequency domain, e.g., cosine domain [Eskicioglu and Fisher, 1995], [Giusto and Perra, 1997], and [Sendashonga and Labeau, 2006], Fourier domain [Mittal *et al*, 1999] and [Ong *et al*, 2004], subbands or wavelets [Lai and Kuo, 2000], [Sendashonga and Labeau, 2006] and [Zhai *et al*, 2005], and Gabor domain [Zhai *et al*, 2006], [Zhai *et al*, 2007], and [Liu and Laganiere, 2007]. These schemes are of moderate complexity and (typically) of slightly improved performance compared to schemes described by (2.11).

Many other QA schemes are reported in the literature, e.g., HVS-based [Daly, 1992], [Miyahara *et al*, 1998], [Algazi *et al*, 1998], and [Ginesu *et al*, 2006], neural networks trained on statistical features the opinion of an expert [Zhou *et al*, 2003], [Brankov *et al*, 2003], [Bouzerdoum *et al*, 2004], information-theoretic (information loss due to some degradation is measured) [Sheikh *et al*, 2005] and [Zhu and Wu, 2005], fuzzy (or fuzzy-II) similarity measures [Weken *et al*, 2002], [Yu and Xie, 2004], [Li *et al*, 2004], and [Weken *et al*, 2007], and finally block-based comparison (using correlation and various statistical moments) [Franti, 1998], [Wang and Bovic, 2002], [Wang *et al*, 2004], [Fernandez *et al*, 2004], and [Wee *et al*, 2007].

Each category has its own advantages and disadvantages as well as its range of application. It should be pointed out that HVS-based objective quality measures are often criticized due to their complexity and the fact that most psychovisual models are based on sinusoidal stimuli.

The above review on QA schemes is to highlight the fact the topic is still in its early stages and more research is needed. Therefore, PSNR, as described by (2.2), will be adopted in this work for comparison purposes. This choice was made since this measure is still widely used in the compression community despite its drawbacks, waiting for a new measure that is simple enough and has better correlation with eye perception.

2.7 Summary

A summary of the most popular compression techniques is given in Table 2.2 below. A general description is only given and the references can be consulted for various variations. Many algorithms were suggested in the literature to combine the benefits (or overcome drawbacks) of two techniques. The author is not aware of an algorithm combining more than two techniques. These algorithms are not stated in Table 2.2.

Table 2.2 Summary of compression techniques.

Technique	Domain	Criterion		Performance
Predictive Coding	pixel-based	Each pixel is predicted from its neighbours, see Fig 2.1, prediction error is quantized and coded		CR~2-4:1. At higher rates, accumulating errors are very annoying
Binary Truncation Coding	block-based	Pixels in the block are divided into 2 groups in a k-means fashion. Two means plus one bit per pixel (indicating group) are sent.		CR cannot exceed 8:1.
Logic Minimization	block-based	Pixel values are treated as binary bits rather than integers. The Boolean function is then minimized and resulting expression is coded.		The only successful implementation was to improve BTC.

Transform Coding	block-based	Each block is considered as a 2D array that is then linearly transformed. The transformed coefficients are quantized and coded.		CR<32:1. DCT is preferred for its energy compaction and simplicity. At higher rates blocking effects are unacceptable.
Quadtrees	block-based	Blocks are split and/or merged according to some similarity measure. Tree information is sent together with a 2D polynomial function describing the block.		Slightly inferior to transform coding. However, can be of simpler implementation.
Triangular meshes	block-based	Similar to the quadtrees technique, however, the blocks are of triangular shape.		Comparable to quadtrees and transform coding in general.
Vector Quantization	block-based	A codebook is generated from a predefined set of images. An index is sent to indicate the block that best match the current block.		Can have higher CR than TC. However, performance is dependant on how similar is the image to the training set.
Wavelet or Subband Coding	whole image	The image is divided into several (usually 4) bands according to frequency contents. 4 filters (2 for each direction) are used to perform this task. The procedure can be applied to the lowest band to obtain a pyramid-like structure.		Performance is superior to that of TC. Complexity is higher though. The resultant image is blurry and is quite annoying for high CR.
Fractal Compression	block-based	Each block is compared to 4 times larger (neighbouring) blocks. Affine transformation parameters are computed for these blocks and applied to a randomly generated image. These parameters are then coded together with distance parameters.		Inferior to wavelet in general but can be better than TC. Its main drawback is the high computational demand since the parameters should be applied for many iterations to reach a good quality.
Region- or segmentation-based Compression	whole image	The image is segmented to several regions depending on some similarity measure or through edge detection. The region is then coded using a 2D polynomial. Boundary pixels should be coded to identify the region from its neighbours.		Performance is expected to be better than other techniques. However, computational demand is still an issue.

Chapter 3

Region Fitting Formulation

This technique shares some aspects with region or segmentation-based compression. Region-based technique is promising for obtaining high compression rates [Sikora 2005]. It has the benefit of decomposing the image into its main objects or subobjects. These objects should be efficiently described. Since the number of objects is quite few in any scene, increased compression ratio is theoretically feasible.

The main drawback (other than the computational complexity), however, is the fact that each region is described by two codes; one represents the contour (due to segmentation) and the second describes the internal colour information of the region enclosed by the contour. This requires two optimization procedures that may be in conflict. In addition, the description of the region may be erroneous depending on the method and information used in the approximation process since segmentation procedures are neither unique nor optimum.

3.1 Region Fitting using Bell-Shaped Functions

A model is proposed to probe further into region-based schemes using a single objective function during optimization. This is a new research trend to overcome the somewhat conflicting demands of combining the coding of the region and its boundary. The traditional way [Muñoz *et al* 2003] is to combine region and boundary information in such a way as to find a better description of one of them with the help of the other. Another goal for the proposed region-fitting framework is that the scheme should lend itself to progressive transmission, i.e., the encoding can continue until a satisfactory quality (lossless can be the ultimate goal) is reached. Partial solution to this problem was proposed [Kaup and Aach 1998] by refining the region content (through DCT) but not its contour. To the author knowledge, no work has been done on any nonpolynomial functions in the region-based image compression context.

In the proposed model, no processing is needed to isolate between high (edges and/or texture) and low (uniform regions) frequency components of the image. Instead, each region is described by a bell-shaped function. Perceptual sensitivity is incorporated for our benefit by avoiding edge extraction and defining the edge as the place of high gradient imposed by the bell function transition region. However, edge strength or sharpness is dependent on how fast the bell function decays. Each region (function) is not restricted to a specific spot in the image but can influence the whole image (with various levels of course). In fact, each function describes the main part of the current region and its boundary and influences the surrounding regions. Mean square error is used, due to its simplicity, as the objective function in the optimization process. This technique requires a huge amount of computation time, however, the theory behind it is quite interesting and requires further investigations. Unfortunately, the author is unable to improve the performance to be competitive in any term to that of most available schemes. Fig 3.1 presents a general description of the model.

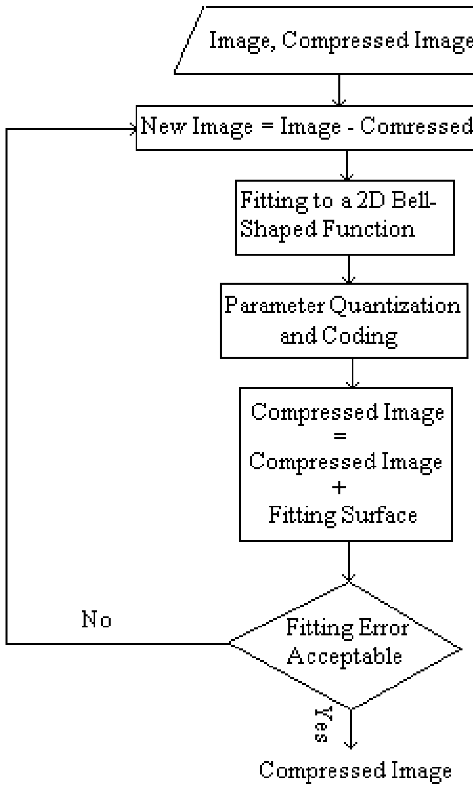

Image, Compressed Image=0
New Image = Image - Comressed Image
Fitting to a 2D Bell-Shaped Function
Parameter Quantization and Coding
Compressed Image = Compressed Image + Fitting Surface
Fitting Error Acceptable
No
Yes
Compressed Image

Fig 3.1 Flow diagram of the region fitting scheme

3.1.1 Mathematical Formulation

The optimization function is similar to that of (2.8). Since each region can be approximated by a slowly varying intensity surrounded by some boundary to separate it from neighbouring regions, it is more economical to consider bell-shaped functions like 2D gaussian-shaped functions for example. Characteristic functions from filter theory like Butterworth and Chebychev share the same definition.

As a first step, a tractable implementation of a bell-shaped function is the first order Butterworth characteristic function, generalized to 2D, resulting in the following optimization problem

$$\underset{a,\,b,\,c,\,x_0,\,y_0}{Minimize} \sum_x \sum_y \left(\sum_i \frac{c_i}{1+a_i^2(x-x_{0i})^2+b_i^2(y-y_{0i})^2} - g(x,y) \right)^2 \qquad (3.1)$$

where $\boldsymbol{i}$ runs over the number of regions, x runs over the horizontal direction, y runs over the vertical direction, $x_{0\mathrm{i}}$ and $y_{0\mathrm{i}}$ are the coordinates of the peak of function i, g is the original image, and a, b, and c are real values characterizing each region. The above equation can be generalized to include regions with different orientation. In this case, the generalized (3.1) can be modified to

$$\underset{a,\,b,\,c\,,x_0\,,y_0\,,\theta}{Minimize} \sum_x \sum_y \left(\sum_i \frac{c_i}{1+A_{xyi}^2(\theta)+B_{xyi}^2(\theta)} - g(x,y) \right)^2 \qquad (3.2)$$

where

$$A_{xyi}(\theta) = a_i\{x\cos(\theta_i) + y\sin(\theta_i) - x_{oi}\}, \text{ and}$$
$$B_{xyi}(\theta) = b_i\{y\cos(\theta_i) - x\sin(\theta_i) - y_{oi}\}$$

Numerical methods are required to solve (3.1) or (3.2). The common derivative-based approaches are Newton-methods (NM) and steepest descent algorithms. The method of Hooke and Jeeves (HJ) [Hooke and Jeeves 1961] is also helpful for derivative free optimization. Genetic algorithms can be incorporated to alleviate trapping in a local minimum. Nevertheless, a more significant problem is that the number of regions is not known apriori.

3.1.2 Iterative Solution

The optimization of (3.2) is computation and memory intensive. As a remedy, successive approximation is used, i.e., (3.2) is solved for one region at a time. The resulting error (difference between the original image and the obtained region) is considered as a new image and (3.2) is applied for the second time. This process continues until meeting the specifications. Though impractical, an exhaustive search mode results are encouraging. Another approach is to obtain seed points, as initial conditions for the parameters in (3.2), through morphological filtering. Some low pass filtering and downsampling may be required before morphological filtering to reduce computational complexity.

The selection of this function, (3.2) with one region, can be helpful in obtaining a direct solution (can be easily reduced to solving a linear system) by taking the reciprocal of both the function and the gray value at each pixel. Similarly, log can be performed when using gaussian functions. Error minimization is then performed on these reciprocals (logs). Other functions can be considered in a similar way. Unfortunately, the solution is not optimum or even appropriate due to the fact that the error at the tail of the bell function effectively receives higher weight than the central part. The last sentence is a clear indication that the solution of (3.2) should be numerical. Newton and quasi-Newton methods are easily trapped in a local minimum. More important is that, for the current optimization problem at least; the Hessian $\nabla^2 f(x_k)$ matrix is ill conditioned for Newton methods. Hence, a steepest descent method like Normalized Least Mean Square [Ameer and Shahrrava 2005] from filter theory may be engaged to avoid such drawbacks on the expense of speed of course.

Genetic algorithm can also be used in the successive optimization of (3.2); however, time consumption is prohibitive. Better results (both in time and quality) are obtained with HJ method (derivative-free direct search algorithm [Hooke and Jeeves 1961]) possibly due to the relatively small number of individuals and generations used in GA. Despite the fact that neither a global minimum (compared to GA) nor improved time performance (compared to Newton methods) is guaranteed, HJ performance and simple implementation makes the model worth exploring.

3.1.3 Experimental Results

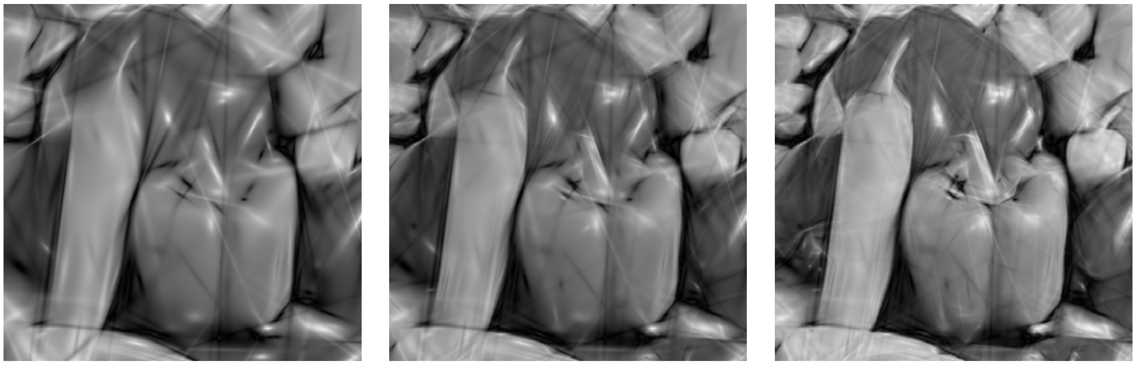


Fig 3.2 Successively approximated image using 256 (left), 512 (middle), 1024 (right) regions at 22.69, 24.61, and 26.52dB respectively.

Fig 3.2 shows three successive approximations of the standard image PEPPER (512x512 with 8 bits per pixel). Region-fitting model is implemented through (3.2) with successive approximation using HJ algorithm. Fifty random initial conditions are used to initiate the optimization routine for each region. The best final result (in term of PSNR) is then selected.

The algorithm is in its early phases and needs improvement, especially with respect to its time complexity. On average, 167 seconds are required (using MatLab 6.5 on a 1.73 GHz centrino processor) to find the 6 parameters of each region. For the 1024 regions, compression ratio is 42.7:1 using 8 bits per parameter. Several runs are needed on different images to get a closer look at the statistics of each parameter and hence implementing an optimum coding scheme at a later stage.

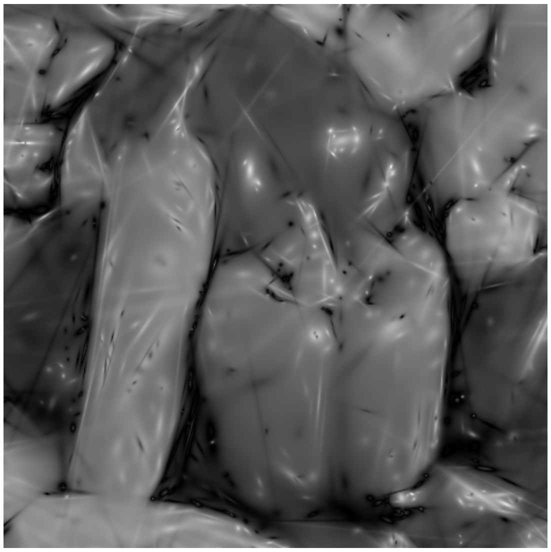


Fig 3.3 Successively approximated image with GA using 1024 regions at 24.18dB.

Fig 3.3 shows the reconstructed image with binary GA using 1024 approximations. The performance is inferior, given parameters and time restrictions, to that of HJ algorithm both in PSNR and time consumption. Each GA approximation (for one region) starts with a random population of 90 individuals that genetically interact for 40 generations. Partial replacement is implemented by selecting the best 90 individuals from the resulting 180 individuals representing all parents and children. Sorting is performed at the end of each generation. Parents' indices are selected according to

$$Parent\ \ index = \left(Number\ \ of\ \ Chromosomes + 1\right)^{\sqrt{random\ \ number}} \qquad (3.3)$$

The rank dependent selection of (3.3), the random number that is generated in the interval (0,1), roughly implements a geometric probability distribution with a geometric ratio of 0.5. Multi-point crossover is used. The crossover positions are randomly selected using a binary random number. Mutation probability is selected as 0.1.

To reduce the computational burden, the initial number of regions and hence their centres is obtained from a plane fitting routine described in chapter 4. After thresholding, the resulting low resolution image has clusters of different sizes. Morphological filtering is then applied to merge and/or split adjacent clusters. Another morphological step is then required to thin or shrink each cluster (region) to a single point. The resulting points represent regions' centres. Although this initialization scheme reduces time consumption but unfortunately decreases the quality. The author relates the failure to the complex surface describing the problem or in other words, the influence of local minima.

3.1.4 Summary and Discussion

A large number of erroneous lines is visible in Figures 3.2 and 3.3. These lines are caused by functions with sharp transitions (possibly due to trapping in a local minimum resulting in a very thin "oval shape"), and to some extent the inadequate representation of negative values. Another observation is that the improvements in PSNR decrease with increased number of regions, and hence, increased computational time.

Unfortunately, Newton and quasi-Newton methods gave inferior results. For Newton methods the reason is mainly due to ill-conditioned Hessian matrix. In many cases, the improvement in PSNR is less than 10^{-5}dB for an added region. Hence, a different approach is needed to overcome this drawback and benefit the fast performance of these methods.

The main advantage of this model is that perceptual quality is inherent by the fact that edges are not explicitly represented but rather treated as an intrinsic component of each region (function). Hence, strong and relevant edges are easily distinguished, even under a low quality implementation as can be seen from Fig 3.2.

The MSE optimization, as its name indicated, aims to fit the largest possible number of points. This can include inhomogeneous pixels in the region and/or may exclude other nearby pixels to fit the required shape. Hence, additional “virtual” regions may be created requiring additional bits and computational time. This, in my opinion, is the main reason behind the failure of this scheme. To our disadvantage, quantization can make the situation even worse. The computational burden; however, is far from being practical due to the high number of iterations required. In addition, one region is found at a time to make the optimization procedure simpler. A preliminary investigation to solve (3.2) successively, using a direct solution, through some mathematical manipulations of (3.2), is proven inadequate.

As previously outlined, the proposed model (unlike region-based techniques) requires a single code to describe the interior and the boundary of a region. Preliminary results indicate that 48 bits are quite enough to describe each region. If compared to the classical way of describing a region with constant intensity (quantized to 5 bits) and an average of 1.5 bits per boundary pixels then 48 bits can only describe a region bounded by about 58 pixels (assuming ~50% of each region boundary is extracted from neighbouring regions). This limitation favours the proposed scheme keeping in mind that with a more efficient coding less than 48 bits can be easily achieved.

3.2 Boolean Regions

A somewhat related approach (to the region fitting scheme) is through Boolean function minimization, i.e., gray values are dealt with as a sequence of ones and zeros not as integers or real values like the previous cases. Implementing Boolean minimization in image compression is generally of limited success. For example, the improvement of BTC using the tabular (Quine-McCluskey) method was suggested in [Augustine *et al* 1999]. Other examples are the lossless compression of binary images and bit maps of gray level images as suggested by [Sarkar 1996] and [Agaian *et al* 2003] respectively.

Attempts have been made to implement different binary transformations and Boolean simplifications on the whole image, as well as blocks of images. Unfortunately, the proposed implementation failed to produce improvement in performance (for both lossy and lossless compression), and at times it even led to expansion rather than compression of the image. The reason could be the lack of an efficient transformation between the binary image and a compact Boolean term. More important is the fact that a Boolean term is generally of rectangular shape (or a group of such shapes) making it difficult to describe arbitrary shapes generally expected in natural images.

Chapter 4

Plane Fitting Formulation

The main objective of this work is to find a more efficient image representation(s) using polynomial fitting. In particular, the first order, i.e. plane, is of special interest due to its computational simplicity. These aspects are pursued in an optimization framework, through a mean square error MSE minimization, to find (or fine tune) the model parameters of these representations. A successful multiplication- and division-free design is implemented to further emphasize on the simplicity issue.

The author is also interested in exploring the deterministic behaviour of image blocks rather than the traditional approach of statistical processes. For example, DCT is considered as having a high compact representation for first order Markov process. In the author's opinion, this is the main reason behind the poor performance of JPEG at high compression rates. A fitting scheme may be more suitable to such a domain as will be demonstrated in the results of this chapter.

The proposed plane fitting scheme is applied to fixed block size image partitioning, however, extension to variable-size blocks may give more freedom in compromising between quality and compression performance. It should be emphasized that implementing variable-size blocks increases the computational complexity substantially. Previous work [Strobach 1991], [Philips 1991], and [Lan *et al* 1998], however, have dealt with quadtree structure of the blocks where a large block is successively divided into 4 subblocks if the fitting error is higher than a threshold.

4.1 General Description

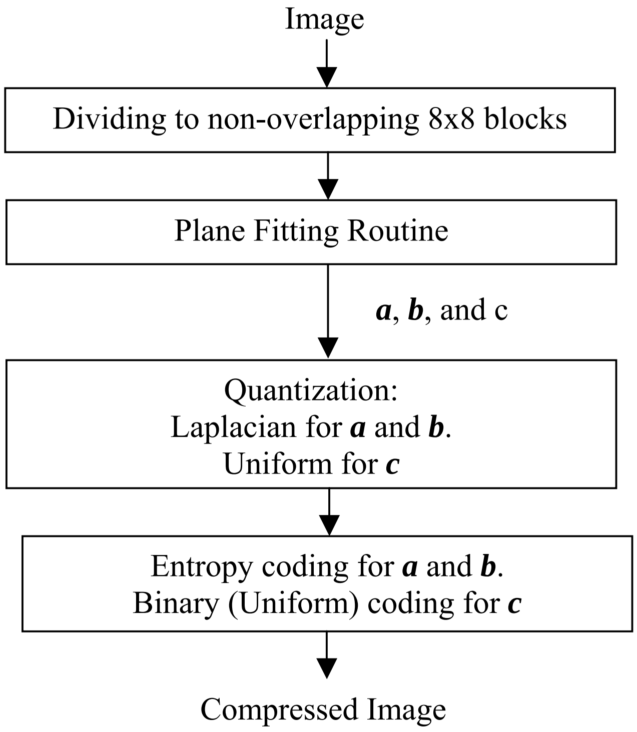


Image
Dividing to non-overlapping 8x8 blocks
Plane Fitting Routine
a, ***b***, and c
Quantization:
Laplacian for ***a*** and ***b***.
Uniform for ***c***
Entropy coding for ***a*** and ***b***.
Binary (Uniform) coding for ***c***
Compressed Image

Fig 4.1 General description of the basic plane fitting scheme.

A plane fitting model is designed to obtain a fast implementation that does not need multiplication or division or, in the worst-case scenario, requires few shifts (see Fig 4.1

for a general description). Although plane fitting is not new to image compression, however, to the author's best knowledge, its application to block-based image compression, using fixed block size, has not been reported.

Simple first order polynomial (plane) fitting [Ameer and Basir 2006] on blocks of size 8x8 pixels demonstrated more than 60:1 compression ratio with acceptable image quality degradation. The results are superior, both in PSNR and visual quality, to that of JPEG at comparable compression ratios. In addition, no multiplications or divisions are required making the implementation suitable for online or progressive compression. Embedded coding is inherent by sending the parameters sequentially.

Despite the simplicity of the model, some features, like the presence of an edge and to some level of accuracy its orientation, can be extracted from the model. This feature could be of great help in enhancing the quality and/or increasing compression ratio. It should be clear that to obtain a better quality, one has to increase polynomial order or decrease the block size and pay the price by decreasing compression ratio. In fact, using a smaller quantization step is of marginal impact on improving quality and will reach saturation after some limit. Blocking effects in the plane model, a shortcoming of block-based compression scheme, are reduced (up to 0.6dB in PSNR improvement) [Ameer and Basir 2006] using simple line fitting at block boundaries. Some manipulations can be performed to reduce the number of computations to about 1.5 additions and 1 shift per boundary pixel.

4.2 Mathematical Formulation

The image is divided into nonoverlapping blocks each is considered as a 3D surface. The z-axis is the pixel value (i.e. the intensity or any colour component). The simplest surface is the plane, i.e., $\boldsymbol{z = ax + by + c}$. To reduce computations, the block centre is selected as the origin. Formulating as a mean square error MSE problem, we have, using (2.9)

$$\underset{a,b,c}{Minimize} \sum_{x=1}^{N} \sum_{y=1}^{N} \{a\, f(x) + b\, f(y) + c - g(x,y)\}^2 \tag{4.1}$$

where *N* is the block dimension; *f(t)* can take one of the following two forms:

$$\begin{aligned} f(t) &= 2t - N - 1 \quad \text{or} \\ f(t) &= \begin{cases} t - N/2 & t > N/2 \\ t - N/2 - 1 & t \le N/2 \end{cases} \end{aligned} \tag{4.2}$$

The first form of *f(t)* was suggested by [Ameer and Basir 2006], however, the second form (N should be even) requires less computations and can lead to slightly better performance. In both forms, a scalable version of each block, and hence of the image, can be easily obtained by normalizing *f(t)*. Setting the derivatives of (4.1) with respect to ***a, b***, and ***c*** to zero results in,

$$a = \mathrm{Z}_{10}, \;\; b = \mathrm{Z}_{01}, \text{and} \;\; c = \mathrm{Z}_{00} \tag{4.3}$$

where,

$$Z_{ij} = \frac{\sum_{x=1}^{N} \sum_{y=1}^{N} f(x)^i f(y)^j g(x,y)}{\sum_{x=1}^{N} \sum_{y=1}^{N} f(x)^{2i} f(y)^{2j}} \tag{4.4}$$

It should be noted that the denominator of (4.4) is independent of the intensity throughout the block and hence can be calculated offline and stored. In fact, some simplification can be done to get, using summation of power series

$$\sum_{x=1}^{N} f(x)^2 = D = \frac{1}{12}\frac{(N+2)!}{(N-1)!} \tag{4.5}$$

To reduce the number of additions, we can sum row-wise (or column-wise) and use the partial sums in finding more than one parameter. Simple manipulations can be performed to convert each multiplication, due to *f(t)*, to 2 shifts or less, e.g., x =3 can be written as x = 2+1 and so on. The number of shifts can be drastically decreased at the decoder by adopting a similar procedure to that of [Hasegawa and Yamasaki 2002] where the intensity of the current pixel is obtained by adding ***a***(***b***) to the horizontally (vertically) preceding pixel.

4.3 Quantization

In previous work on plane fitting [Strobach 1991] and [Lan *et al* 1998], uniform quantization was adopted for the three plane parameters ***a*** , ***b*** and ***c***. However, extensive experiments on several natural images have revealed that the distribution of ***c*** is uniform, while, the distributions of ***a*** and ***b*** are similar and each can be approximated by a zero mean random variable having a nonuniform probability distribution of the form $e^{-k\sqrt{x}}$ approximately. Leading to the fact that quantization should be nonuniform for ***a*** and ***b*** but uniform for ***c***. It has also been found that if the upper left corner of the block was selected as the origin, the range of ***c*** would increase by more than 20%, and hence more bits are required (or lower quality for the same bit budget), compared to the case of selecting the block centre as the origin. In fact, the density function is approximate and hence the results of (2.3) may not be accurate. Therefore, a direct search optimization routine is implemented, on some test images, to find quantization levels (apart from the zero value) given by

$$L(q,n) = (-1)^n \, 26.5 \frac{e^{2.2q/Q} - 1}{e^{2.2} - 1} \tag{4.6}$$

where q=0, 1, …, Q–1; Q is the number of quantization levels, and n=0,1 represents right and left tails of the distribution respectively. Each value of ***a*** and ***b*** is quantized to the nearest quantization level in the above equation. The values 26.5 and 2.2 were found empirically to give a good compromise between CR and PSNR. Changing these two values over a reasonably wide range produces different compromise between CR and PSNR. However, these differences in performance are marginal on average for less than 15% range (image dependent). The pre-computed levels in (4.6) are of great help in eliminating the division required by (4.4).

4.4 Encoding

To eliminate the need for a second processing pass and/or sending coding tables, a Huffman table is constructed from measurements on several images. By adopting this scheme, suboptimal performance is traded for simplicity. The table has $(2Q+1)^2$

elements, each representing a code for the block compound symbol (***a***,***b***). The (measured) probabilities are forced to be symmetric, through simple averaging, with respect to the (0,0) element.

It is noted that the probability of occurrence is inversely proportional to the absolute value of the parameter. As expected, the code is smaller for smaller values. A binary coding of 5 bits long is used to encode ***c***. This encoding scheme results in a slightly better CR performance compared to that proposed by [Ameer and Basir 2006] where comma coding was implemented.

4.5 Postprocessing

At the decoder, block boundaries (both horizontally and vertically) are linearly interpolated to reduce blocking effects. [Ameer and Basir 2006] implemented a procedure that ignores pixel values at the blocks' boundaries and replaces them with those obtained from interpolating the two neighbouring pixels, exactly one pixel to the right (top) and left (down) for horizontal (vertical) smoothing.

A more efficient procedure is implemented that uses first order curve fitting on four pixels, two from each block, to find better estimates of the boundary pixels. Mathematically,

$$\begin{aligned} \hat{g}(Nx, y) &= r - m \\ \hat{g}(Nx + 1, y) &= r + m \end{aligned} \tag{4.7}$$

where,

$m = v * k' / (v * v')$,
$k = [\hat{g}(Nx-1, y)\ \hat{g}(Nx, y)\ \hat{g}(Nx+1, y)\ \hat{g}(Nx+2, y)]$,
$v = [-4 \quad -1 \quad 1 \quad 4]$, and
$r = \sum k / 4.$

A similar procedure is applied to the vertical direction, i.e.,

$$\begin{aligned} \hat{g}(x, Ny) &= r - m \\ \hat{g}(x, Ny+1) &= r + m \end{aligned} \tag{4.8}$$

where $k = [\hat{g}(x, Ny-1)\ \hat{g}(x, Ny)\ \hat{g}(x, Ny+1)\ \hat{g}(x, Ny+2)]$ and the rest of the parameters are the same as those in (4.7). Simple additions and shifts are required in (4.7) and (4.8), except for the division (to find *m*) that can be eliminated (for 8x8 blocks) with the following modification

$$m = v * k' / 32 \tag{4.9}$$

and hence, a division is replaced by a simple shift right operation.

4.6 Multiplication- and Division-Free Implementation

Due to the simplicity of the computation of the parameters in (4.3), the multiplication- and division-free implementation is straightforward by first rounding the values of (4.6) to the nearest integer. However, a more elaborate description is required when the prediction scheme is incorporated as will be described in the next chapter.

For Q=4, we have, ±14, ±6, ±2, and 0. These values together with those of $f(t)$ in (4.2) are easily manipulated to a multiplication- and division-free version. The division in (4.4) can be easily overcome by merging it with the calculation of the quantization levels. Extending the procedure to other values of Q or a different block size is almost trivial.

The number of encoder calculations required per block, for blocks of size 8x8, to find all row- and column-wise partial sums is 112 additions. Seven more additions are required to find $\boldsymbol{c}$. Finding $\boldsymbol{a}$ and $\boldsymbol{b}$, on the other hand, requires 10 additions and 8 shifts. In the decoder, however, 63 additions are required at most. The actual number of additions is less than 63, since the zero value is the most probable one. In any case, the higher complexity is in the encoder where 129 additions and 8 shifts are required per block. This is much less than the multiplierless implementation of JPEG proposed by [Tran 2000] where 480 additions and 240 shifts are required per block. Keeping in mind that the multiplierless implementation of [Tran 2000] is less efficient than the original JPEG version. In [Chan and Lee 2006], an algorithm is proposed to overcome the quality degradation resulting from [Tran 2000]. However, the number of computations is larger by around 25%.

4.7 Experimental Results

Results are demonstrated on the standard image PEPPER (512x512 with 8 bits per pixel) shown in Fig 4.2. Fig 4.3 illustrates the reconstructed image before and after linear interpolation, described by (4.7), using 8x8 blocks, Q=4 and 5 bits to encode $\boldsymbol{c}$. It is clear that uniform regions are well described with tolerable edge degradations. For a closer look at the reconstruction error, Fig 4.4 shows zooming of the images shown in Fig 4.2 and Fig 4.3.

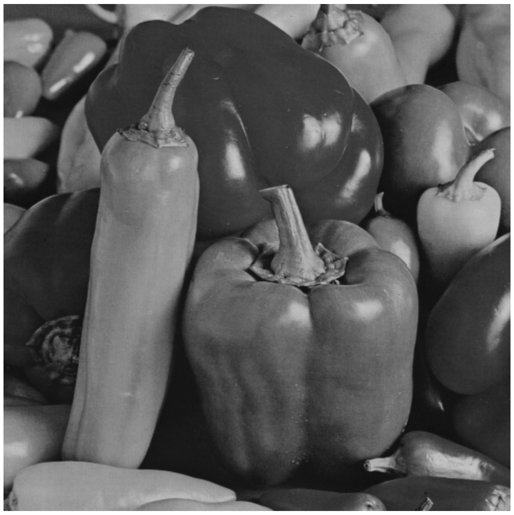


Fig 4.2 Original image PEPPER

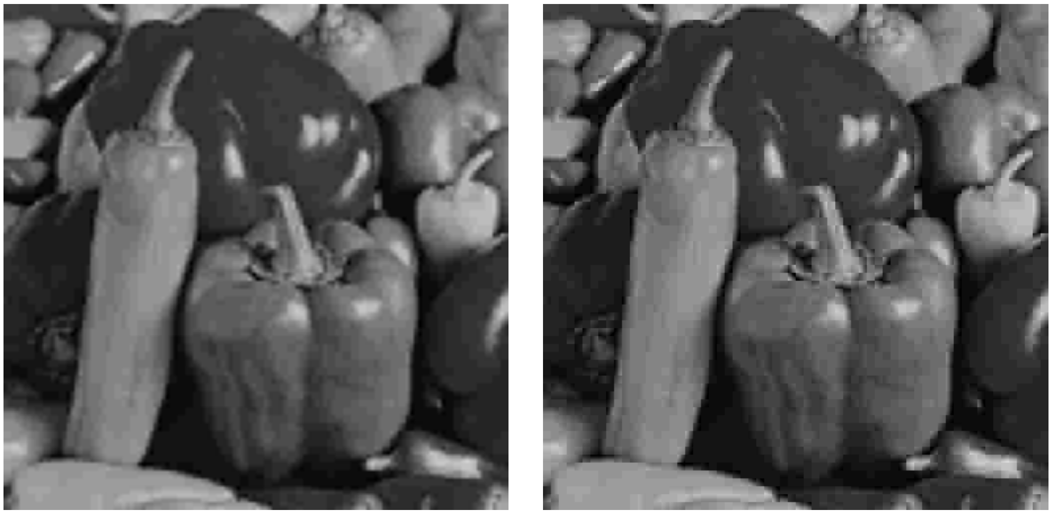


Fig 4.3 Reconstructed images at 62:1 compression using 8x8 blocks (left) with boundary interpolation at 28.8dB and (right) no interpolation at 28.2dB.

In comparison, JPEG image, obtained from the MatLab command *imwrite*, is shown in Fig 4.5. The proposed scheme is better in CR by about 35% compared to JPEG. Although PSNRs are comparable (around 28.8dB), visual quality of the reconstructed image using the plane fitting scheme is more pleasing than that of JPEG.

In fact, the proposed scheme is visually more pleasing than JPEG even without interpolation at blocks' boundary.

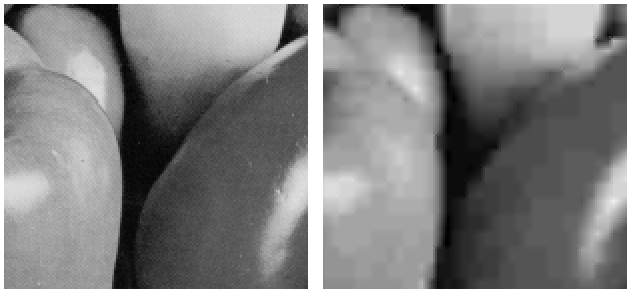


Fig 4.4 Zooming of (left) original and (right) reconstructed (with linear interpolation) images.

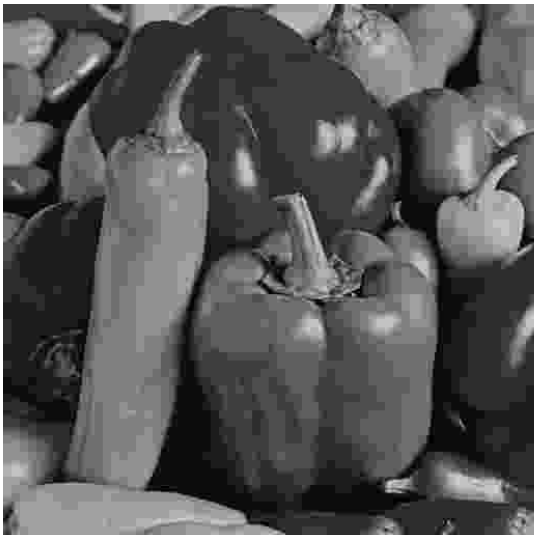


Fig 4.5 JPEG image at 46.01:1 compression and 28.76dB.

Additional 2 bits are needed per image to send the value of *Q* (2 – 5) and another 2 bits to indicate the number of quantization bits for ***c*** (3 – 6). Using the proposed uniform quantization of [Strobach 1991] and [Lan *et al* 1998] was inferior as was expected since the data collected by the author clearly indicates the nonuniform distribution of ***a*** and ***b***. It is worth mentioning that the proposed quantization, for the parameters ***a*** and ***b***, reduces PSNR by less than 0.3dB compared to the no quantization case. The proposed scheme is implemented with different block sizes and the results are presented in Fig 4.6. The results are comparable (superior in some cases) to those listed in [Biswas 2003]. It is interesting to note that even values of *N* have higher CR than *N* – 1 with slight reduction in PSNR. This is mainly related to the presence of zero in *f(t)* for odd N values, see (4.2). Hence, one row and one column will not contribute to the evaluation of the parameters.

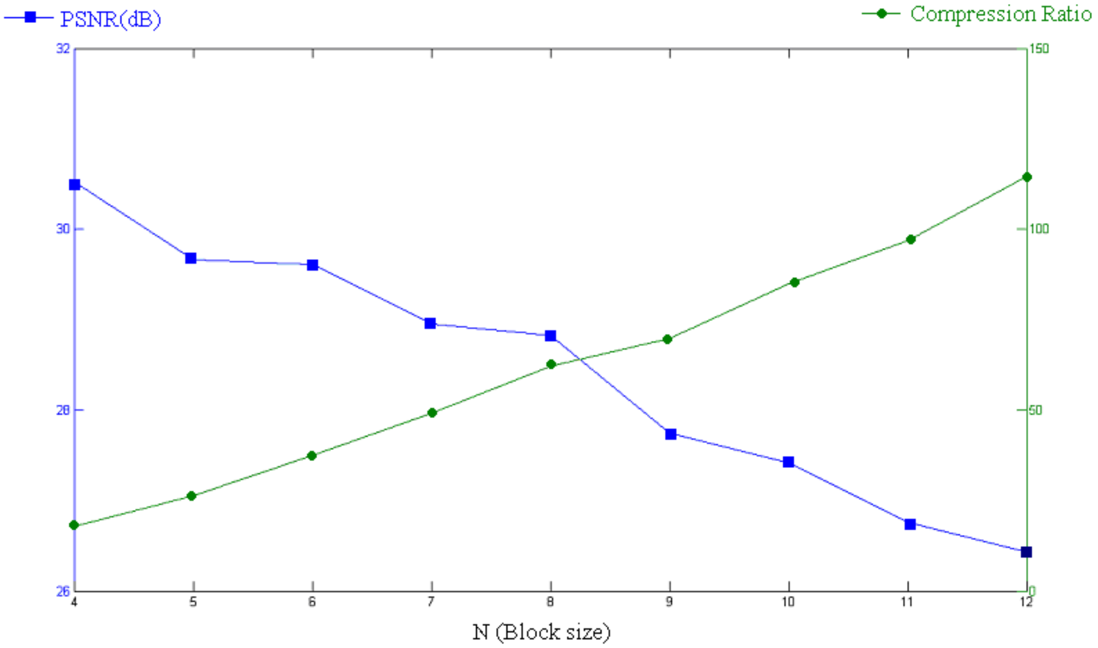


Fig 4.6 Performance for different block sizes (Q = max(2, N/2))

For completeness, Fig 4.7 illustrates the performance results of implementing overlapped blocks by one pixel from each side. In a similar fashion, Table 4.1 shows results for different block sizes overlapped by *N* – 8 pixels. These results demonstrate the marginal PSNR gain obtained at the expense of substantial CR reduction when using overlapped blocks. Performance of the proposed plane fitting scheme, before and after the postprocessing of (4.7), on other standard images is listed in Table 4.2 together with some results implementing the JPEG standard. It is noted that the proposed scheme is superior and in some cases JPEG cannot reach the proposed CR. It is worth mentioning that in all cases of Table 4.2, though comparable PSNR is obtained, the visual quality of JPEG is inferior.

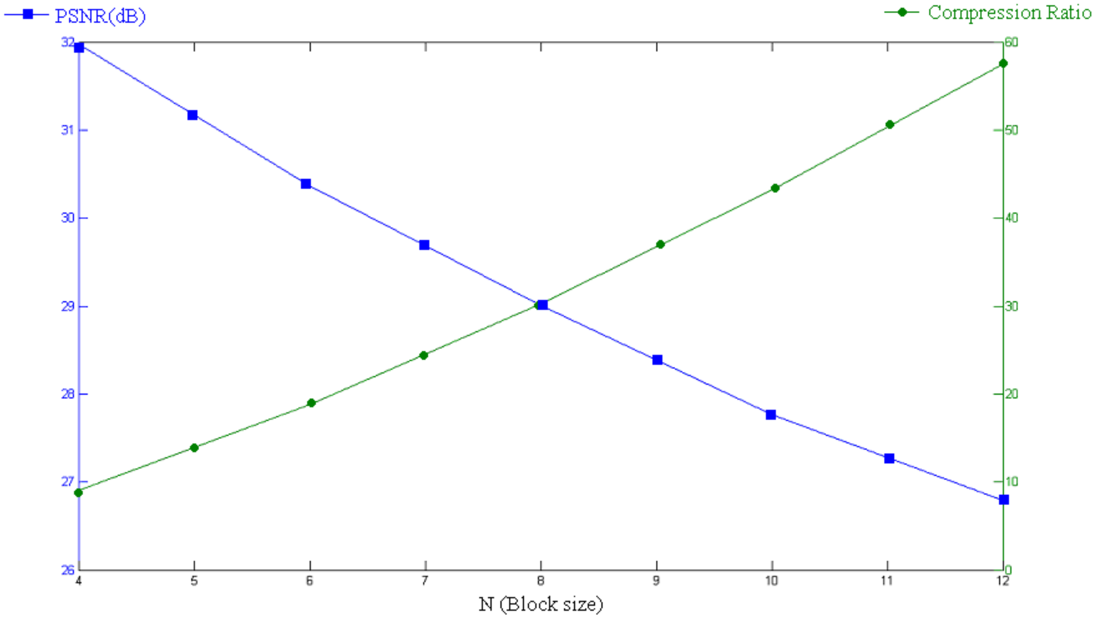


Fig 4.7 Performance for different overlapped block sizes

Table 4.1 Performance for different sizes overlapped by N – 8 pixels

N	**PSNR (dB)**	**CR**
9	25.79	54.29
10	25.79	53.63
11	25.40	51.08

Table 4.2 Performance on different images using 8x8 blocks.

JPEG with a similar proposed PSNR	JPEG with a similar proposed CR	***Image***	Proposed		
			PSNR (dB)		CR
PSNR/CR	PSNR/CR		No Processing	With Processing	
23.87/50.80	---	***Balloon***	23.66	23.97	61.13
26.03/49.21	23.96/59.76	***Boat***	25.86	26.45	60.91
26.21/46.43	24.00/58.37	***Chimp***	26.05	26.42	58.80
28.48/50.28	---	***Fern***	28.43	29.28	67.61
20.79/37.89	---	***Mandrill***	20.58	20.67	54.95
23.78/43.45	---	***Temple***	23.19	23.56	56.95

An image dependent compensation of around 0.5dB is obtained with linear interpolation at block boundaries; however, the visual quality is not improved that much for *N*>8. No significant differences were noticed between the implementations of (4.7) and (4.9). Interpolation gains are higher for odd *N* than for even *N*. Diagonal interpolation, after horizontal and vertical interpolation, produces an insignificant degradation of 0.03dB. A negligible improvement of 0.02dB is obtained with a one step extrapolation in the four directions of each block. In fact, ignoring the boundary pixels in the fitting process can increase compression ratio to 63.99:1 at the same PSNR. However, subjective quality is marginally degraded. A better reduction of blocking effects (around 0.7dB) was obtained with 10-point cubic fitting. This slight increase did not improve visual quality and is not favoured against the linear interpolation due to the excessive computation required.

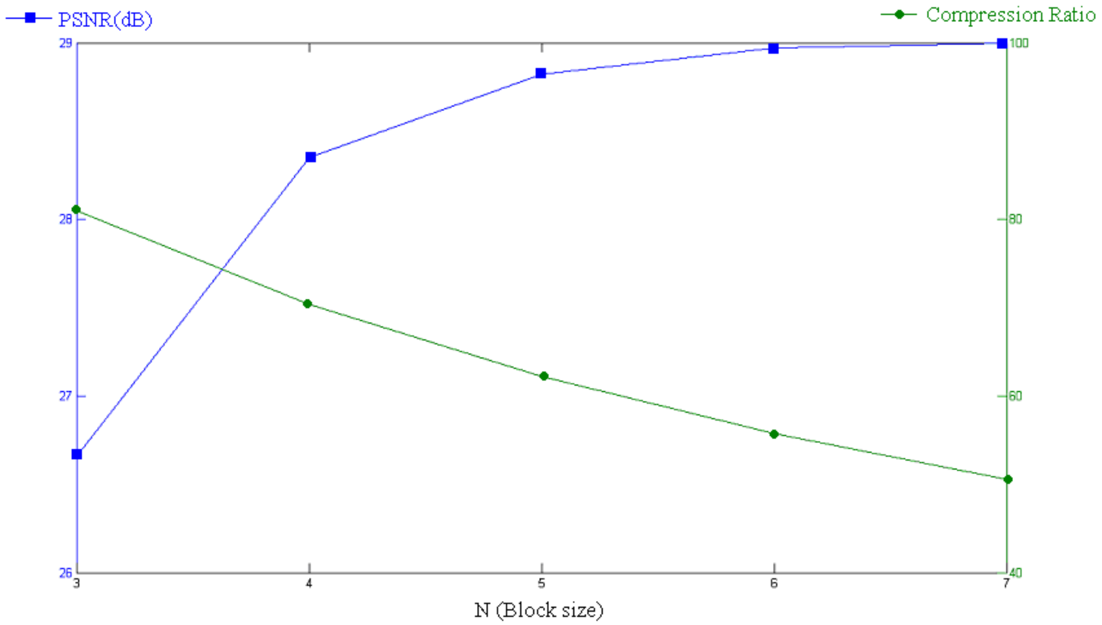


Fig 4.8 Performance for different quantization of ***c*** on image PEPPER after processing (*Q*=4).

PSNR improvement of 0.1dB (at 55.78:1 compression) can be obtained when quantizing ***c*** to 6 bits as demonstrated in Fig 4.8. This slight increase is visually more pleasing especially in homogeneous regions. In fact, the reconstruction quality is sensitive to the quantization of ***c*** more than to that of ***a*** and ***b***. Fig 4.9 shows some results for different values of *Q* when ***c*** is quantized to 5 bits. As for subjective quality, reconstructed images are visually acceptable; however, the cases *Q* = 2 and *Q* = 3 are slightly bothersome because of blockiness. No significant differences are noticed between other values of *Q*. Around 10% increase in CR is obtained (PSNR decreases by 0.2dB) when sending a quantized ***(a + b)/2*** and ***(a – b)/2*** instead of ***a*** and ***b***. However, the visual quality is similar to that of *Q* = 3 (see Fig 4.9).

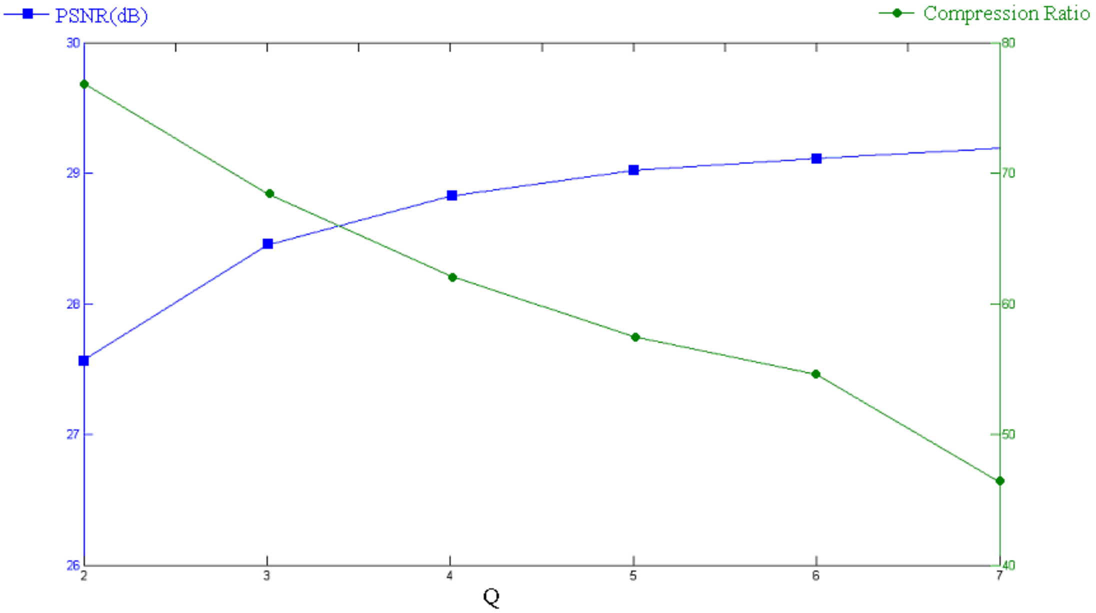


Fig 4.9 Performance for different *Q* on image PEPPER using 8x8 blocks (after postprocessing and quantizing ***c*** to 5 bits).

4.8 Summary and Discussion

A simple block-based compression scheme is proposed using plane fitting. The proposed scheme is novel in its application to fixed size blocks. In addition, multiplierless implementation has slight influence on the overall performance. An MSE formulation is optimized to obtain the three plane parameters. These parameters are then quantized and coded. The quantized values have different probabilities and hence, a Huffman code is constructed to improve coding efficiency. However, uniform distributions were assumed in previous models [Strobach, 1991]. To further improve compression rate, the table is constructed such that each code represents a symbol representing the two values (***a***, ***b***) rather than sending a code for ***a*** and another one for ***b***. The proposed scheme is computationally attractive since it is working in a multiplication- and division-free mode. To maintain simplicity, postprocessing is also performed in a multiplication- and division-free environment.

Superiority to JPEG, both objectively and subjectively, has been demonstrated for compression ratios >50:1. The proposed scheme, however, needs to be improved to maintain its superiority for low compression ratio applications. The author believes that the improvement should be in the form of increasing the order, reducing block size, or a combination of that. Stated differently, three parameters are not enough to describe fine details of image blocks.

It should be emphasized that high values of ***a*** (***b***) clearly indicate the presence of horizontal (vertical) edges. This observation is justified by the fact that for small blocks the intensity is almost constant unless there is an edge. The orientation of the edge can be inferred from the magnitude of the ratio between ***a*** and ***b***. However, these indicators fail to estimate the location of the edge. In fact, $\boldsymbol{a}x+\boldsymbol{b}y+\boldsymbol{k}$, $\boldsymbol{k}$ is to be determined, is a line fairly approximate the edge contour. A more serious problem, although rare in practice, occurs when there are multiple edges in the same block. These edges can either went unpredicted or erroneously predicted as a single edge. Fuzzy set theory may be of great help in this regard to reach an improved solution. In addition, the inherent information inside $\boldsymbol{a}$ and/or $\boldsymbol{b}$ can be useful in postprocessing or in finding a more efficient coding by sharing the same information with neighbouring blocks. Since errors are usually larger at edges than inside uniform regions, the previous discussion is worthy to be further investigated.

Chapter 5

Higher Order Polynomial Fitting Formulations

The promising results achieved in the plane model motivate the author to investigate the implementation of higher order polynomials in block-based image compression. Obviously, increasing the polynomial order improves the quality of the resultant image. The price paid is the reduction in compression ratio. The objective of this chapter is to investigate some higher order polynomials in block-based image compression. The investigation is constrained with the requirement that compression ratios should be comparable to that of the plane model in its basic form. The higher order polynomial schemes have higher computational burden and inferior performance compared to that of the plane fitting scheme.

It is difficult to implement polynomial schemes with orders greater than two. Hence, some insight in an approximate asymptotic behaviour is performed using a new linear mapping scheme. The results of this chapter clearly indicate the power inherent in the plane model. Nevertheless, the proposed new trends in higher order polynomial may open the way for future improvements when used alone or mixed with other schemes.

5.1 Separable Monotonics

In this case, in a similar fashion to the plane fitting setting in chapter 4,

$$z = a\ sign(x)\ |x|^{m} + b\ sign(y)\ |y|^{n} + c \tag{5.1}$$

Minimizing MSE, we have

$$\begin{aligned} a &= \sum_{x=-(N-1)/2}^{(N-1)/2} \sum_{y=-(N-1)/2}^{(N-1)/2} \frac{x|x|^{m-1}\, g(x,y)}{N S_x} \\ b &= \sum_{x=-(N-1)/2}^{(N-1)/2} \sum_{y=-(N-1)/2}^{(N-1)/2} \frac{y|y|^{n-1}\, g(x,y)}{N S_y} \\ c &= Z_{00} \end{aligned} \tag{5.2}$$

where Z_{00} is as defined in (4.4), $S_x = \sum |x|^{2m}$ and $S_y = \sum |y|^{2n}$. The best MSE performance is for the plane case, i.e., $m = n = 1$.

5.2 Adding the *xy* Term

Here we have ***z=(ax + c) (by + c)***. Minimizing MSE we get [Ameer and Basir 2006]

$$\begin{aligned} &a/c = k + \sqrt{k^2 + r},\ c^2 = \frac{Z_{10}a/c + Z_{00}}{(a/c)^2 + 1} \\ &\text{and} \qquad bc = \frac{Z_{11}a/c + Z_{01}}{(a/c)^2 + 1} \end{aligned} \tag{5.3}$$

where

$$k = \frac{Z_{11}^2 - Z_{01}^2 - Z_{00}}{2Z_{11}Z_{01}}, \text{and} \quad r = \frac{Z_{10} + Z_{11}Z_{01}}{Z_{11}Z_{01}} \tag{5.4}$$

The solution of (5.3) is obtained through a two-stage optimization. First, the solution for ***z=d'xy + a'x + b'y + c'*** is found. Following that, a comparison is made between the primed and unprimed parameters to find the unprimed ones. ***a*** and ***b*** follow their plane counter parts in (4.6) but with different quantization parameters. The above solution is suboptimal. A better solution can be obtained using the form *z =* ***c (a*** *f(x)+1)(****b*** *f(y)+1)* in (2.8) and minimizing MSE, through a single optimization step, we get

$$\begin{aligned} a &= k + sign(r)\sqrt{k^2 + N\Big/\sum_{j=1}^{N} f(x_j)^2} \\ c &= \frac{aZ_{10} + Z_{00}}{N\left(a^2\sum_{j=1}^{N} f(x_j)^2 + N\right)} \\ b &= \frac{aZ_{11} + Z_{01}}{aZ_{10} + Z_{00}} \frac{N}{\sum_{j=1}^{N} f(x_j)^2} \end{aligned} \tag{5.5}$$

where

$$
\begin{aligned}
r &= NZ_{11}Z_{01} + Z_{10}Z_{00}\sum_{j=1}^{N} f(x_j)^2 \\
2rk &= \frac{N^2 Z_{11}^2}{\sum_{j=1}^{N} f(x_j)^2} + N\left(Z_{10}^2 - Z_{01}^2\right) - Z_{00}\sum_{j=1}^{N} f(x_j)^2
\end{aligned}
\tag{5.6}
$$

The above equations are valid for *$r \neq 0$* and *$ar+N(Z_{01}^2 - Z_{10}^2)>0$*. If ***NOT*** then we have

$$
\begin{aligned}
c &= Z_{00} \big/ N^2 \\
a &= \begin{cases} 0 & Z_{10} < Z_{01} \\ \dfrac{NZ_{10}}{Z_{00}\sum_{j=1}^{N} f(x_j)^2} & Z_{10} \geq Z_{01} \end{cases} \\
b &= \begin{cases} 0 & Z_{10} \geq Z_{01} \\ \dfrac{NZ_{01}}{Z_{00}\sum_{j=1}^{N} f(x_j)^2} & Z_{10} < Z_{01} \end{cases}
\end{aligned}
\tag{5.7}
$$

In a similar fashion to (4.6), due to fact that the density function is approximate, a direct search optimization routine is implemented to find quantization levels of ***a*** and ***b*** given by

$$
L(q,n) = (-1)^n 0.5 \frac{e^{3.3q/Q} - 1}{e^{3.3} - 1} \tag{5.8}
$$

q and *n* are as defined in (4.6). A better quality is obtained when quantizing ***a*** and ***b*** after being multiplied by ***c*** and the quantization levels are modified to

$$
L(q,n) = (-1)^n 25.4 \frac{e^{2.1q/Q} - 1}{e^{2.1} - 1} \tag{5.9}
$$

The four-parameter case, i.e., ***z=dxy + ax + by + c*** has better PSNR than the plane case. However, it will not be considered further due to the increased complexity (by ~33%) and the great reduction in CR.

5.2.1 Experimental Results

Fig 5.1 shows the reconstructed image for the xy case. The performance is almost the same as that of the plane case. However, the complexity is much higher and hence the plane case is favoured.

Similar to the plane case, the quantization of ***c*** has more influence on the subjective quality of the reconstructed image. This is not surprising since ***c*** represents the average gray level of the block.

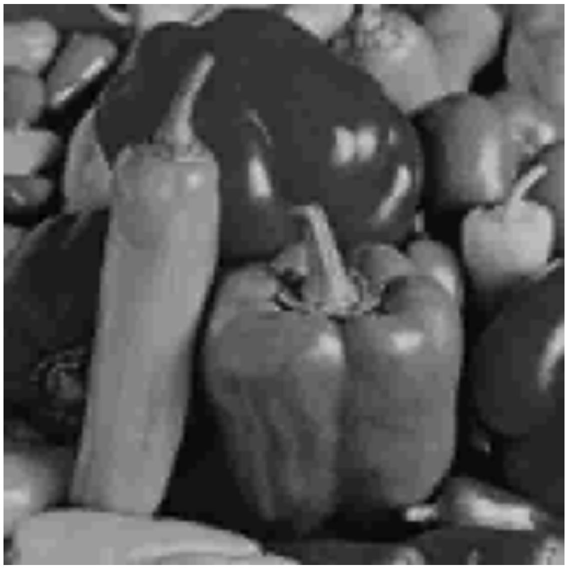


Fig 5.1 Reconstructed image (xy case) at CR=62.17and 28.82dB with ***c*** quantized to 5 bits and Q=4.

5.3 Quadratic Surface

Different combinations of three unknowns are tried, e.g., $\boldsymbol{z=(ax + by + c)^2}$ and $\boldsymbol{z=(ax + c)^2 + (by + c)^2}$. The solutions should be obtained through nonlinear equations requiring numerical methods. Even though the performance is poor and hence was not pursued further. The discussion in the previous section also applies here for the six-parameter case, i.e., $z=dx^2 + ey^2 + fxy + ax + by + c$ and also for the influence of ***c***.

5.4 Higher Orders

Many surfaces can be fitted, using three unknowns, by gray scale transformations of the form ***f[g(x, y)]***, i.e.

$$\underset{a,b,c}{Minimize} \sum_{x=-(N-1)/2}^{(N-1)/2} \sum_{y=-(N-1)/2}^{(N-1)/2} \left[ax + by + c - f\left(g(x,y)\right)\right]^2 \qquad (5.10)$$

Different forms of ***f(x)*** were tested. However, quality is inferior to that of the plane case for many functions like exponentials, logarithms, trigonometric, and powers of order different than one. When $f(x) = x^r$ in (5.10) the performance reaches its optimum at $r = 1$.

5.5 Linear Mapping of Parametrically Generated Features

In this section a new linear mapping scheme for image compression is proposed. It can be considered as an approximate asymptotic behaviour to higher order polynomials. Each block of the image is independently reconstructed from a set of “features” through a linear mapping. Fig 5.2 shows a block diagram of the proposed structure [Ameer and Basir 2008a].

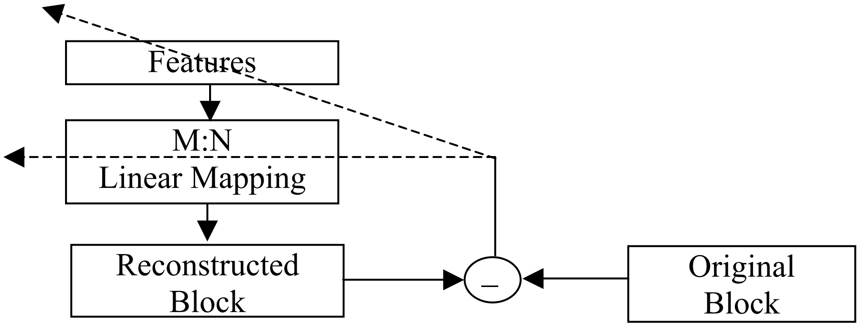


Fig 5.2 Block diagram of the proposed linear mapping structure.

Inspired by the fact that independent (or at least uncorrelated) features result in an efficient and compact description of the patterns, a random sequence generator is employed, using a sine function with two parameters, to generate a set of "uncorrelated" features. These two parameters and the set of weights between the features, generated by the sine function, are obtained through a training process. The outputs are the reconstructed values of the image block. A simple description of the system is presented in Fig 5.3.

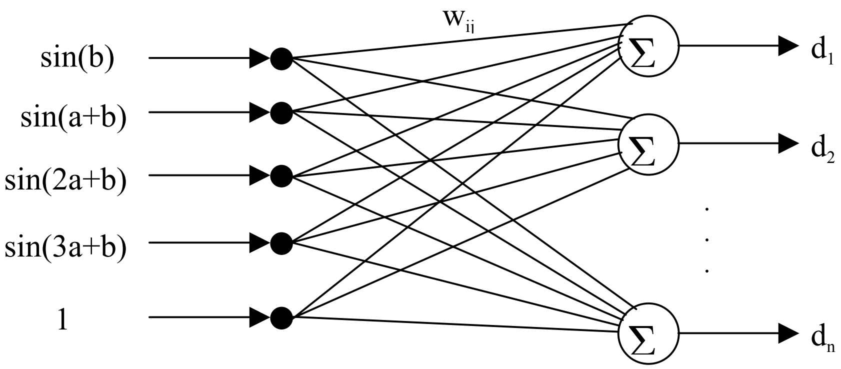


Fig 5.3 The proposed structure using 4 features.

The first pass of the optimization process [Ameer and Basir 2008a] is performed off-line by iteratively solving a set of linear systems together with a steepest descent scheme to find the weights and the sine function parameters respectively. Only the weights are then made available for both transmission ends. To perform compression on any image not necessarily in the training set, the sine function parameters should be found on-line for the compressed image and then quantized. The computation time is excessive due to the nonlinear optimization (steepest descent) procedure required. However, thanks to quantization, a look-up table (a parallel implementation is even better) can be a good alternative to overcome this disadvantage.

The main idea is inspired by the fact that optimal performance in the feature subspace is obtained when features are maximally independent. Fortunately, independency and uncorrelation are equivalent for Gaussian processes. For non-Gaussian processes, however, the performance is sub-optimal. The proposed structure is fairly independent of the random sequence length as experimentally demonstrated. The easiest and most primitive way to generate a set of "uncorrelated" values is through a random number generator. Since all trigonometric functions are modulo 2π, a parameterized sine function is used to replace modulo operation. Hence, the "features" in the proposed model are *sin(aj+b)* where *j* runs from 1 to *m*, the number of features. The proposed structure, shown in Fig 5.3 using four such features, has some similarity with neural networks' structure; however, the underlying theory is different as will be described in the following sub-sections.

5.5.1 Off-Line Phase

In this phase, the training image is divided into non-overlapping blocks that are sequentially used by the proposed structure as its optimum or desired response. The objective function to be minimized can be set accordingly as [Ameer and Basir 2008a]

$$s=\sum_{k=1}^{P}\sum_{i=1}^{N}\left\{\sum_{j=0}^{m-1}\left\{w_{ij}\sin\left(a_k j+b_k\right)\right\}+w_{im}-d_{ik}\right\}^2 \tag{5.11}$$

where P is the number of training patterns (number of nonoverlapping image blocks in this case), N is the number of pixels in the block, m is the number of “uncorrelated” features, to be described later, w_{im} is the weight connecting output node i to the bias unit m, w_{ij} is the weight connecting output node i to feature j, and d_{ik} is the i^{th} pixel value in block k after suppressing the block mean.

The first step in finding the optimum parameters of (5.11) is by setting all derivatives to zero, hence

$$\frac{\partial s}{\partial a_k}=0=\sum_{i=1}^{N}\left\{\sum_{j=0}^{m-1}w_{ij}\sin\left(a_k j+b_k\right)+w_{im}-d_{ik}\right\}\left\{\sum_{u=0}^{m-1}uw_{iu}\cos\left(a_k u+b_k\right)\right\} \tag{5.12}$$

$$\frac{\partial s}{\partial b_k}=0=\sum_{i=1}^{N}\left\{\sum_{j=0}^{m-1}w_{ij}\sin\left(a_k j+b_k\right)+w_{im}-d_{ik}\right\}\left\{\sum_{u=0}^{m-1}w_{iu}\cos\left(a_k u+b_k\right)\right\} \tag{5.13}$$

$$\frac{\partial s}{\partial w_{it}}=0=\sum_{k=1}^{P}h_{tk}\left\{\sum_{j=0}^{m-1}w_{ij}\sin\left(a_k j+b_k\right)+w_{im}-d_{ik}\right\},\quad h_{tk}=\begin{cases}1 & t=m\\ \sin\left(a_k t+b_k\right) & t<m\end{cases} \tag{5.14}$$

The problem is obviously non-linear in ***a*** and ***b*** and linear in ***w***. Therefore, each optimization pass has two steps. In the first step ***a*** and ***b*** are updated using a steepest descent algorithm with a learning rate of 10^{-6}. In the second step, the N linear systems of (5.14) are solved for ***w*** to find the updated weights. The learning process continues until some convergence criterion is met. Finally, the weights are made available for both communication ends. The off-line phase of the proposed structure is not fast. However, the time-wise cost is negligible since it is done only once. It should be emphasised that the values of ***a*** and ***b*** found in this phase are irrelevant and are only updated to get an optimum solution for ***w***.

5.5.2 On-Line Phase

In this phase, to compress an image not necessarily from the training set, the same setup of Fig 5.3 is used except that ***w*** is now known from the off-line phase. ***a*** and ***b*** for each block are found by iteratively solving (5.12) and (5.13). Following that, ***a*** and ***b*** are quantized and sent to the receiver which then reconstruct an approximation of each block of the original image using the same model of Fig 5.3.

An approximate distribution of ***a*** (***b***) is built using the values obtained from 9 different images. A direct search scheme is then implemented to find the optimum quantization levels of ***a*** and was found as (apart from level 1) [Ameer and Basir 2008a]

$$1 \pm 0.68 \frac{e^{2.7i/4} - 1}{e^{2.7} - 1}, \qquad i = 1,2,3 \tag{5.15}$$

Similarly, the levels of ***b*** are (apart from level 0)

$$\pm 0.68 \frac{e^{2.4i/3} - 1}{e^{2.4} - 1}, \qquad i = 1,2 \tag{5.16}$$

In both (5.15) and (5.16), changing the parameters on a fairly wide range has marginal effects on the performance. The above quantization schemes are of great help in reducing the computation time and hence, solving (5.12) and (5.13) can be replaced by choosing the best performance, in satisfying (5.11), obtained from all of 35 possible combinations of ***a*** and ***b***. Computation time can be further reduced by performing the 35 computations in parallel. A Huffman coding table can now be built. However, to further increase CR, a Huffman coding table is built using the occurrence of the compound symbol (***a***,***b***) taken from a large set of images.

It should be kept in mind that PSNR could be increased (<1dB depending on the image) using some post-processing scheme like the ones proposed in chapter 4 or later in chapter 6 of this dissertation for example.

5.5.3 Experimental Results

The image is divided into non-overlapped blocks of size 8x8 pixels. Training (off-line) stops when the total error of the current epoch, one sequential pass through all blocks, is not less than 0.999 of the error in the previous one. Several trials are then performed with different initial conditions to choose the weights with best quality performance.

The proposed structure is implemented with m=4 and marginal differences were noticed for 3≤m≤6. Performance began to fluctuate afterwards within 0–3 dB from that obtained with m=4. The block mean is quantized to five bits and sent as overhead information. Results for some images are demonstrated in Table 5.1. As expected, quality is better for the training image. No post-processing is used in this simulation.

Table 5.1 Performance of 512x512 images using 8x8 blocks (trained on PEPPER) using the proposed model (m=4).

Image	SNR	PSNR	CR
Barbara	12.9769	24.2209	67.1411
Boat	20.8609	25.7389	63.7878
Cameraman	17.6673	25.8839	68.9082
Chimp	21.2812	25.9126	62.6520
Fern	17.2120	28.3090	69.6150
Lena	21.4130	27.0169	65.0582
Mandrill	15.3467	20.5376	58.5388
Pepper	19.5146	27.9436	65.1310
Temple	15.9994	23.1138	60.0507

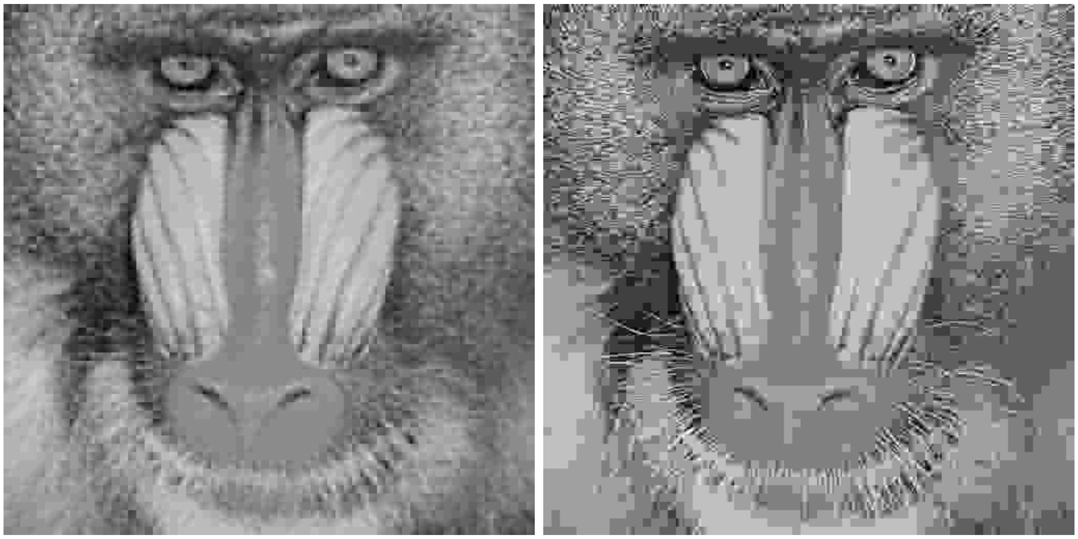


Fig 5.4 (left) Reconstructed Mandrill at CR=58.5 and PSNR=20.5dB and (right) JPEG version at CR=46.6 and PSNR=20.1dB.

Mandrill and its JPEG version are presented in Fig 5.4 for comparison. The same comparison is done for PEPPER (training image) in Fig 5.5. Clearly the proposed algorithm outperforms JPEG both objectively and subjectively. On the other hand, although the comparison is “unfair’, the objective performance of JPEG2000 (being not a block-based scheme) is less by 2–5dB compared to the proposed model but subjectively “more pleasing”.

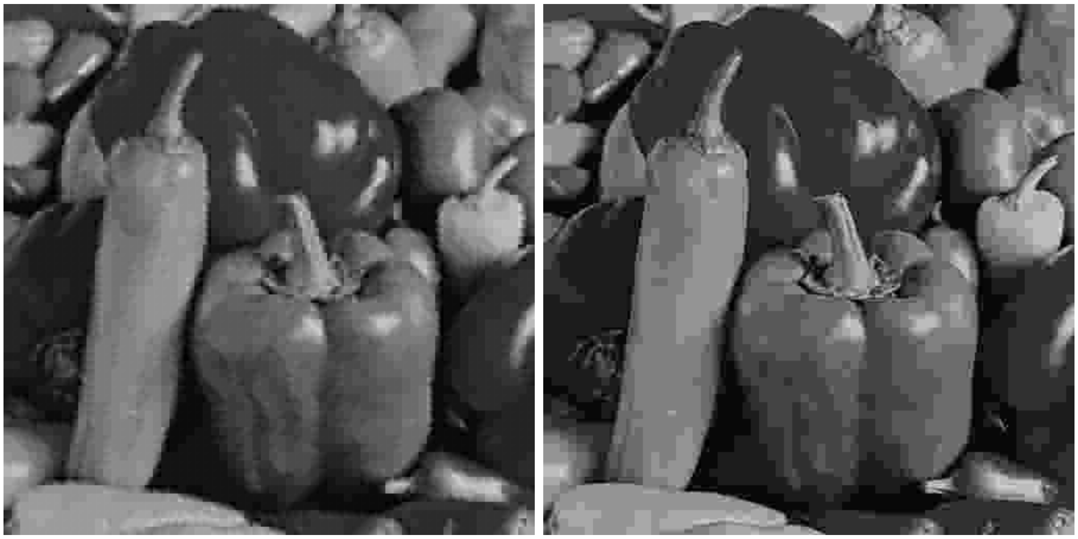


Fig 5.5 (left) Reconstructed PEPPER at CR=65.1 and PSNR=27.9dB and (right) JPEG version at CR=49.5 and PSNR=27.9dB.

As a comparison with neural network schemes, the performance reported by [Costa and Fiori 2001], CR=64 and SNR=18.09dB for Lena (training image), is less by more than 3.3dB from that given in Table 4.1. The proposed structure also outperforms that of [Ma and Khorasani 2002], Fig(11) therein, even though Lena is the training image there.

5.6 Summary and Discussion

Higher order polynomial models (upto second order) are investigated and compared with the plane model proposed in chapter 4. To maintain the compression performance gained in the plane model, all other models are restricted to three parameters. With this setup, the plane model is superior in terms of quality and simple implementation.

In order to get a better insight on the ultimate behaviour of higher order models, a linear mapping scheme is proposed. Parametrically generated features are linearly mapped to approximate the current block. Again, only three free parameters are used. This framework has some similarity with neural networks and hence has similar universal approximation power of neural networks. The plane model is highly comparable to the linear mapping scheme, emphasizing the power inherit in the plane model despite its simplicity.

Chapter 6

Plane Fitting with Inter-Block Prediction

As can be anticipated from Chapter 4, a great portion of the bit budget is used to code ***c*** in the basic plane fitting scheme. Hence, it would be beneficial to find a better representation of ***c*** to increase CR. It is noticed that the range of ***c*** decreases with increasing |***a***|+|***b***|. Nevertheless, performance differences of such implementation were not substantial (<5% increase in CR with 0.5dB decrease in PSNR).

One idea that has been explored by other researchers (JPEG standard for example) is to predict ***c*** from the ***c***'s of neighbouring blocks [Furht 1995]. The author, however; has found that this scheme results in poor performance when combined with the basic plane fitting scheme. As a remedy to this situation, the author proposed a novel approach to predict ***c*** using the intensity value of a certain pixel on the boundary of one of the neighbouring blocks. This specific pixel (or its modified value) is optimally selected through a simple comparison step as will be described in the following sections.

It is assumed throughout this chapter that a post-processing step given by (4.9) or the modification in (4.10) is always performed to improve the quality of the reconstructed image. A second postprocessing stage is then implemented to further improve the quality. As will be experimentally demonstrated, the second postprocessing stage is of subjective rather than objective characteristics mainly due to the incorporation of Weber law. To the author's knowledge, no work has been reported on such an implementation of Weber law. As a by-product, an edge detection scheme is proposed using a simple modification to Weber law. The proposed edge detection scheme is a new trend in combining perceptual laws in the detection process. Interestingly, the same law can be easily modified to act as a quality assessment scheme emphasising the universality of Weber law application in image processing.

6.1 Unoptimized Prediction of *c*

A pixel in a neighbouring block is used to predict ***c*** according to [Ameer and Basir, 2009]

$$c_i = \hat{g}_i(u_i, v_i) - af(x_i) - bf(y_i) \tag{6.1}$$

where ***a*** and ***b*** are as in (4.3), $\boldsymbol{c}_\mathrm{i}$ is the predicted value of ***c*** according to neighbouring block *i*, *i* is the index of the causal neighbouring block as illustrated in Fig 6.1, $\hat{g}_i$ is the matrix representing the intensities in reconstructed neighbouring block *i*, and x_i, y_i, u_i, and v_i are as shown in Table 6.1.

```
. . . . . . . . . . . . . . .
. . . . 1  2  3 . . . . . .
. . . . 0  x  . . . . . . .
. . . . . . . . . . . . . . .
```

Fig 6.1 Causal neighbourhood of block x

Table 6.1 Parameters' definition for (6.1).

i	***0***	***1***	***2***	***3***
x_i	N/2+1	1	1	1
y_i	1	1	N/2+1	1
u_i	N/2+1	N	N	N
v_i	N	N	N/2+1	1

Obviously, some of the choices in (6.1) are omitted for blocks at image boundary. Comparison between the alternatives in (6.1) is then made to select the neighbour that results in the lowest error. The lowest error can be obtained by simplifying (4.1) and substituting (4.4), resulting in

$$error_i = (Nc_i)^2 + N(a^2 + b^2)\sum_{j=1}^{N} f(x_j)^2 - 2(c_i Z_{00} + aZ_{10} + bZ_{01}) \qquad (6.2)$$

The parameters are as defined in (4.2) and (4.4). CR will increase substantially since an index of less than two bits (on average) is used to replace the five-bit code used for ***c*** in the basic scheme. In this case, quantization and encoding schemes are the same as those of the basic scheme (for ***a*** and ***b***) with the addition of the coding of the index of the neighbouring block that best approximate the value of ***c***. The probability of occurrence of these indices is not the same. It is experimentally found that the probabilities are such that $P_0 \geq P_2 > P_1 \geq P_3$ (see Fig 6.1). Hence, an image-independent Huffman table is constructed to code these indices.

Higher CR (but lower PSNR) can be obtained by restricting the neighbourhood to blocks 0 and 2.

6.2 Prediction Constrained Optimization

The performance reported in the previous section can be further improved by substituting (6.1) in (4.1) and performing an optimization step with respect to the remaining two variables, namely ***a*** and ***b***, resulting in [Ameer and Basir, 2009]

$$\begin{aligned} a_i &= \frac{Z_{10} + f(x_i)\left\{N^2 \hat{g}_i(u_i, v_i) - Z_{00}\right\} - Nhf(y_i)}{N\left\{D + Nx_i^2 + Ny_i^2\right\}} \\ b_i &= \left\{a_i f(y_i) + h/N\right\}/f(x_i) \\ \text{where} \quad h &= \left\{Z_{01} f(x_i) - Z_{10} f(y_i)\right\}/D \end{aligned} \tag{6.3}$$

where Table 6.1 is used, *D* is as defined in (4.5), and the rest are defined in (4.2) and (4.4). The four solutions given by (6.3) are then compared to select the one with the lowest error given by (6.2). Again, higher CR (but lower PSNR) can be obtained by restricting the neighbourhood to blocks 0 and 2 as in Fig 6.1.

Index encoding is the same as in the previous section. Quantization levels follow a similar pattern to that of (4.6). Again since the density function is approximate, a direct search optimization routine is implemented to find quantization levels of ***a*** and ***b*** given by

$$L(q,n) = (-1)^n 30 \frac{e^{3q/Q} - 1}{e^3 - 1} \tag{6.4}$$

where q=0, 1, …, Q–1; Q is the number of quantization levels, and n=0,1 represents right and left tails of the distribution respectively. Each value of $\boldsymbol{a}$ and $\boldsymbol{b}$ is quantized to the nearest quantization level in the above equation. The values 30 and 3 were found empirically to give a good compromise between CR and PSNR. Changing these two values over a wide range does not produce significant changes in the performance on average. The pre-computed levels in (6.4) are of great help in eliminating the division required by (6.3).

6.3 Comments on Fine-tuning

The pixel position used to predict $\boldsymbol{c}$ from neighbouring blocks 1 and 3 is the right and left lower corner respectively. For neighbouring blocks 0 and 2, however, the optimum pixel position to predict $\boldsymbol{c}$ is the $(N/2+1)^{\text{st}}$ pixel counted from the top-left corner. In fact, a slightly better performance can be obtained using the average of $(N/2+1)^{\text{st}}$ and $(N/2)^{\text{th}}$. An attempt was made to predict $\boldsymbol{c}$ from other combinations but no significant improvements were noticed.

Marginal improvements can be obtained using higher order prediction models on the expense of complex computation and more important lower compression ratio. Predicting any two parameters, from the four neighbouring blocks, was of poor quality. Marginal degradations were noticed when the prediction is performed using an extrapolated version of each neighbouring block.

The compression ratio can be further increased by ignoring the index bit if the predicted value of ***c*** is the same from all neighbouring blocks using the quantized values of ***a*** and ***b***. A more aggressive increase can be obtained by considering the fuzzy meaning of the word "same", however, the quality degradation may be objectionable.

It is worth mentioning that a second processing step can be performed, if time and complexity are not critical, to design a specific coding table for each image, and perhaps, an adaptive quantization scheme, with a slight increase in overhead bits.

6.4 Encoder Multiplierless Implementation

In this section, a multiplierless implementation of (6.2) and (6.3) is presented for the case of two neighbours. We have

$$x_1 = y_2 = 0, \quad x_2 = y_1 = p = f(1), \quad \mathrm{r} = 2\sum_{i=1}^{n} i^2$$

and

$$\begin{aligned} h_0 &= \frac{\hat{g}_0(N/2, N) + \hat{g}_0(N/2+1, N)}{2} \\ h_2 &= \frac{\hat{g}_2(N, N/2) + \hat{g}_2(N, N/2+1)}{2} \end{aligned} \tag{6.5}$$

Hence, the ***a***, ***b***, and ***c*** parameters are given by

$$a_0\left(nr + n^2p^2\right) = Z_{10} + np^2 Z_{10}/r \Rightarrow a_0 = \frac{Z_{10}}{nr} \tag{6.6}$$

$$b_2\left(nr + n^2p^2\right) = Z_{01} + np^2 Z_{01}/r \Rightarrow b_2 = \frac{Z_{01}}{nr} \tag{6.7}$$

$$a_2\left(nr + n^2p^2\right) = Z_{10} + p\left(n^2h_2 - Z_{00}\right) \tag{6.8}$$

$$b_0\left(nr + n^2 p^2\right) = Z_{01} + p\left(n^2 h_0 - Z_{00}\right) \tag{6.9}$$

$$\begin{aligned} c_0 &= h_0 - b_0 p \\ c_2 &= h_2 - a_2 p \end{aligned} \tag{6.10}$$

The error of (6.2) can now be rewritten as

$$S_i = \left(nc_i\right)^2 + nr\left(a_i^2 + b_i^2\right) - 2\left(Z_{00}c_i + Z_{10}a_i + Z_{01}b_i\right) \tag{6.11}$$

For prediction from the left neighbour, (6.11) reduces to

$$S_0 = n^2 h_0^2 - 2Z_{00}h_0 - \frac{Z_{10}^2}{nr} - \frac{\left\{Z_{01} + p\left(n^2 h_0 - Z_{00}\right)\right\}^2}{nr + n^2 p^2} \tag{6.12}$$

Multiplying by (nr+n^2p^2) and adding the term $Z_{10}^2 + Z_{01}^2 + p^2 Z_{00}^2$, we have

$$S_0 = n^3 r h_0^2 - 2nrZ_{00}h_0 - \frac{Z_{10}^2 np^2}{r} - 2pZ_{01}\left(n^2 h_0 - Z_{00}\right) \tag{6.13}$$

Multiplying by n/r and adding Z_{00}^2, we get

$$S_0 = \left(n^2 h_0 - Z_{00}\right)^2 - \frac{Z_{10}^2 n^2 p^2}{r^2} - \frac{2npZ_{01}\left(n^2 h_0 - Z_{00}\right)}{r} \tag{6.14}$$

Ignoring constants and rearranging, we get

$$S_0 = \left|n^2 h_0 - Z_{00} - npZ_{01}/r\right|^2 \tag{6.15}$$

The power of 2 in (6.15) can be dropped to simplify comparison. Following the same steps for prediction from the top neighbour we get

$$S_2 = \left|n^2 h_2 - Z_{00} - npZ_{10}/r\right| \tag{6.16}$$

If p=–$n/2$ and using discrete integration we have

$$r/2=\sum_{i=1}^{-p}\left[i(i+1)-i\right] = \frac{-p(-p+1)(-2p+1)}{6} \tag{6.17}$$

Restricting ourselves to *n=2^m*, we have,

$$r=\frac{n(n+1)(n+2)}{12} \tag{6.18}$$

$$S_0=\left|n^2h_0-Z_{00}+\frac{6nZ_{01}}{(n+1)(n+2)}\right| \tag{6.19}$$

$$S_2=\left|n^2h_2-Z_{00}+\frac{6nZ_{10}}{(n+1)(n+2)}\right| \tag{6.20}$$

For *n*=8, Equations (6.19) and (6.20) can be approximated by

$$S_0\cong\left|64h_0-Z_{00}+\frac{Z_{01}}{2}\left(1+\frac{1}{16}\right)\right| \tag{6.21}$$

$$S_2\cong\left|64h_2-Z_{00}+\frac{Z_{10}}{2}\left(1+\frac{1}{16}\right)\right| \tag{6.22}$$

Let O_a, O_m, and O_s are the costs of addition, multiplication, and one-bit shifting operations respectively, the number of operations per block required in the encoder is:

1. Row- and column-wise sums require 2n(n-1) O_a.
2. Computing Z_{00} requires (n-1) O_a.
3. Computing Z_{10} and Z_{01} requires
   - 2{(O_a+2 O_s)+(2 O_a+O_s)+(O_a+O_s)+(O_a)+(3 O_a)}, for n=8.
   - 2{(O_a+O_s)+(O_a)+(O_a)}, for n=4.
   - 2(O_a), for n=2.
4. Computing Z_{10} and Z_{01} terms in S_0 and S_2 requires

- $2(O_a+5\ O_s)$, for n=8.
- $2(O_a+2\ O_s)$, for n=4.
- No operation, for n=2.

5. Computing h_0 and h_2 requires $2(O_a+O_s)$.
6. The rest of S_0 and S_2 require
   - $2(2\ O_a+6\ O_s)$, for n=8.
   - $2(2\ O_a+4\ O_s)$, for n=4.
   - $2(2\ O_a+2\ O_s)$, for n=2.
7. 1 comparison is required to choose S_0 or S_2.
8. The parameters ***a*** and ***b*** are then found using (6.6) and (6.9) or (6.7) and (6.8) depending on the outcome of (6.2). Assuming we have two quantization tables, one for each variable, we can avoid division. However, $(O_a+ O_s)$ are required for implementing (6.8) or (6.9).
9. Finding the ***c***'s require
   - $2(O_a+2\ O_s)$, for n=8.
   - $2(O_a+O_s)$, for n=4.
   - $2\ O_a$, for n=2.
10. 1comparison for the index bit.

In total,

- *for n=8, $\sim N^2\{2.31\ O_a+0.57\ O_s\}$ are required per image.*
- *for n=4, $\sim N^2\{2.88\ O_a+1.19\ O_s\}$ are required per image.*
- *for n=2, $\sim N^2\{4.5\ O_a+1.75\ O_s\}$ are required per image.*

For JPEG2000, however, assuming lossy Daubechies (9,7) filter on 5 levels and one scaling factor, the number of computations required is $\{4(2\ O_a+2\ O_m)+2\ O_m\}\ 2\ \{N^2+ (N/2)^2+ (N/4)^2+ (N/8)^2+ (N/16)^2\}/2 \sim N^2\{10.66\ O_a+13.33\ O_m\}$. This is far more than the proposed scheme even for multiplierless implementations replacing the filter coefficients with its nearest power of two. In fact, a multiplierless implementation (assuming 5 levels) of lossy Daubechies (5,3) filter, inferior by 1 dB to floating point arithmetic [Abbas and Tran 2006], requires $(8\ O_a+13\ O_s)2N^2\Sigma^{2}_{i=0}(1/4)^i/2 \sim N^2\{10.66\ O_a+17.33\ O_s\}$ which is more than five times the complexity of the proposed scheme.

Interestingly, the proposed scheme, as will be shown in the results section, is superior to the JPEG. However, a multiplierless implementation of the DCT suggested by [Tran 2000] requires $N^2\{7.5\ O_a +3.25\ O_s\}$ operations that are also higher than that of the proposed.

6.5 Decoder Multiplierless Implementation

At the decoder, computation complexity is mainly due to postprocessing. Two stages of postprocessing are implemented. The linear fitting scheme of (4.7) is applied first followed by a Weber-based postprocessing scheme. Weber law has been shown to have a good correlation with the human visual system [Pratt, 2001]. A detailed description of the modified Weber law is proposed in Chapter 7. The number of operations required by the decoder is shown below for the case n=8. Generalizing to other values of n is straightforward. The computational requirements are as follows:

1. Computing ***c***, 2(2 O_a+2 O_s) operations are required plus 1 comparison. The proper ***c*** is then uploaded in all cells of the block.
2. Starting from the pixel (1,1) we need to perform n(n-1)O_a of the smallest absolute and (n-1)O_a of the largest. Due to the fact that most of the added values are zeros, the number of additions may be a small portion of n^2–1. This can be further reduced if we have enough memory to store all the possibilities.
3. For the block boundary enhancement, Equation (4.7), 2(N/n-1)N{(3 O_a+2 O_s)+(3 O_a+2 O_s)+(5 O_s)+(2 O_a)} operations are required.
4. For Weber-based postprocessing (Chapter 7), the difference is found between the points next to the end points of all horizontal, vertical, and oblique directions (e.g., points a and b or c and d or a and e or b and d or d and e in Fig 7.1) and multiplied by
   - [1/2] for 1 pixel,
   - [1/4+1/16, 1–(1/4+1/16)] for 2 pixels,
   - [1/4, 1/2, 1–1/4] for 3 pixels,
   - [1/8+1/16, 1/2–1/8, 1/2+1/8, 1–(1/8+1/16)] for 4 pixels,
   - [1/8+1/16, 1/4+1/16, 1/2, 1–(1/4+1/16), 1–(1/8+1/16)] for 5 pixels,
   - [1/8, 1/4+1/16, 1/2–1/16, 1/2+1/16, 1–(1/4+1/16), 1–1/8] for 6 pixels,
   - [1/8, 1/4, 1/2–1/8, 1/2, 1/2+1/8, 1–1/4, 1–1/8] for 7 pixels, and
   - [1/8, 1/4, 1/4+1/16, 1/2–1/16, 1/2+1/16, 1/2+1/8, 1–1/4, 1–1/8] for 8 pixels.

In all 8 cases, except for the case of 1 point, line end points, e.g. points α and β in Fig 7.1, are averaged and the average is then subtracted from points a and b. The same calculations are repeated for each direction (46 in total). Mathematically, we find max(|(α+β)/2–a|, |(α+β)/2–b|) and compare to max{(α+β)/2, 255–(α+β)/2}/16. The only way to perform postprocessing is when comparison indicates a less than or equal status. Hence, the required number of operations is $(O_a+O_s)(42)+2\ O_a(46)+O_a(46)+46(2\ O_a+4\ O_s)+O_a(46)$ just to do the required comparisons. In addition, interpolation requires (assuming 50% true comparisons) $222\ O_a+80\ O_s$. In fact, it is 97% of that due to boundary blocks.

Totally, the number of operations is $\sim N^2\{11.22\ O_a+6.92\ O_s\}$. The complexity could be less than this since, for horizontal or vertical lines, we skip comparison if the index bit indicates a prediction from top or left neighbour respectively. Nevertheless this is less complex than JPEG2000, using Daubechies (9,7) with each multiplication replaced by a single addition and shifting, see the last two paragraphs in the previous section.

6.6 Experimental Results

The results of Sections 6.1 and 6.2 are implemented using MatLab 6.5 and are given in Table 6.2. The table clearly indicates the increase in quality at the expense of minor reduction in compression ratio. In addition, the result of CR for two neighbours with optimization can be increased by 7% if we do not send the index bit when the ***c***'s are the "same", see (6.1).

Table 6.2 Performance of the proposed inter-block prediction scheme for image PEPPER (see text for computational aspects).

Performance / Method	4 Neighbours		2 Neighbours	
	PSNR	CR	PSNR	CR
Without Optimization	27.45	103.01	26.68	121.43
With Optimization	28.38	90.32	27.92	105.49

It is noticed that the degradation due to the multiplierless implementation does not exceed 0.2dB for the images reported in Table 4.2. In the software implementation, using MatLab, no significant difference in speed was noticed between the direct and the multiplierless implementations. However, theoretically at least, the difference in cost is approximately equal to the ratio between the costs of addition and multiplication in the platform used for implementation.

As a visual comparison for two cases from Table 6.2, Fig 6.2 illustrates two reconstructed versions of the standard image PEPPER. Interestingly, the performance is much better than sending the block average (***c***) only where CR=102.4 and PSNR=24.1 dB. A multiplier- and division-free implementation incorporating the Weber-based postprocessing, described later in Chapter 7, is shown in Fig 6.3. More details are given in the next chapter on the implementations of Weber-based postprocessing with other image compression schemes.

An objective comparison between the proposed scheme and JPEG2000 reveals the superiority of the proposed scheme in terms of CR, PSNR, and computational complexity. Stated differently, more than 5dB in PSNR (1.3 times in CR) with complexity cost described in sections 6.4 and 6.5 for the encoder and decoder respectively.

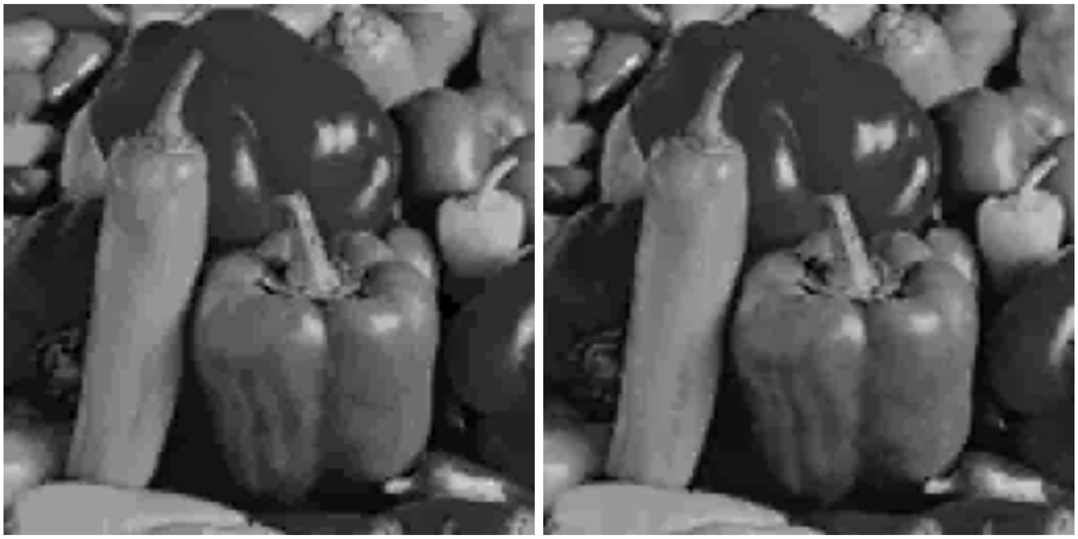


Fig 6.2 Reconstructed images, (left) 4-point prediction without optimization at CR=103.0 and 27.45dB, and (right) 2-point prediction with optimization at CR=105.5 and 27.9dB.

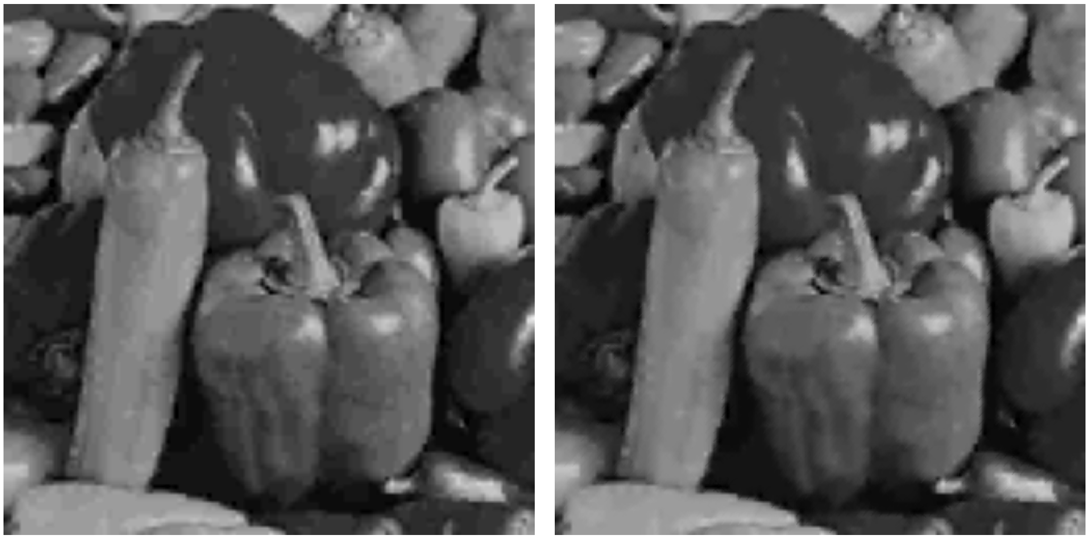


Fig 6.3 Reconstructed images (multiplierless version with optimized 2-point prediction) at CR=114.51, (left) before postprocessing PSNR=27.73dB, and (right) after postprocessing PSNR=27.75dB.

To perform subjective quality comparison, the opinions of 22 viewers on 4 quality aspects were recorded. Four images were presented to the viewers each image is accompanied by its two compressed versions using the proposed scheme and JPEG2000 as shown in Fig 6.4. Method (M1) is the proposed scheme and (M2) is JPEG2000 using similar CR slightly greater than 100:1. The viewers do not know which is which and were asked to indicate which of the reconstructed images is closer to the original in terms of edges, objects' interior uniformity, illumination, and information preserving. It can be inferred from the viewers' statistics in Table 6.3, that the proposed scheme is subjectively more pleasing than JPEG2000 for compression ratios >100:1 especially for edge preserving aspect. However, as can be deduced from Table 6.3, the comparison is image dependant.

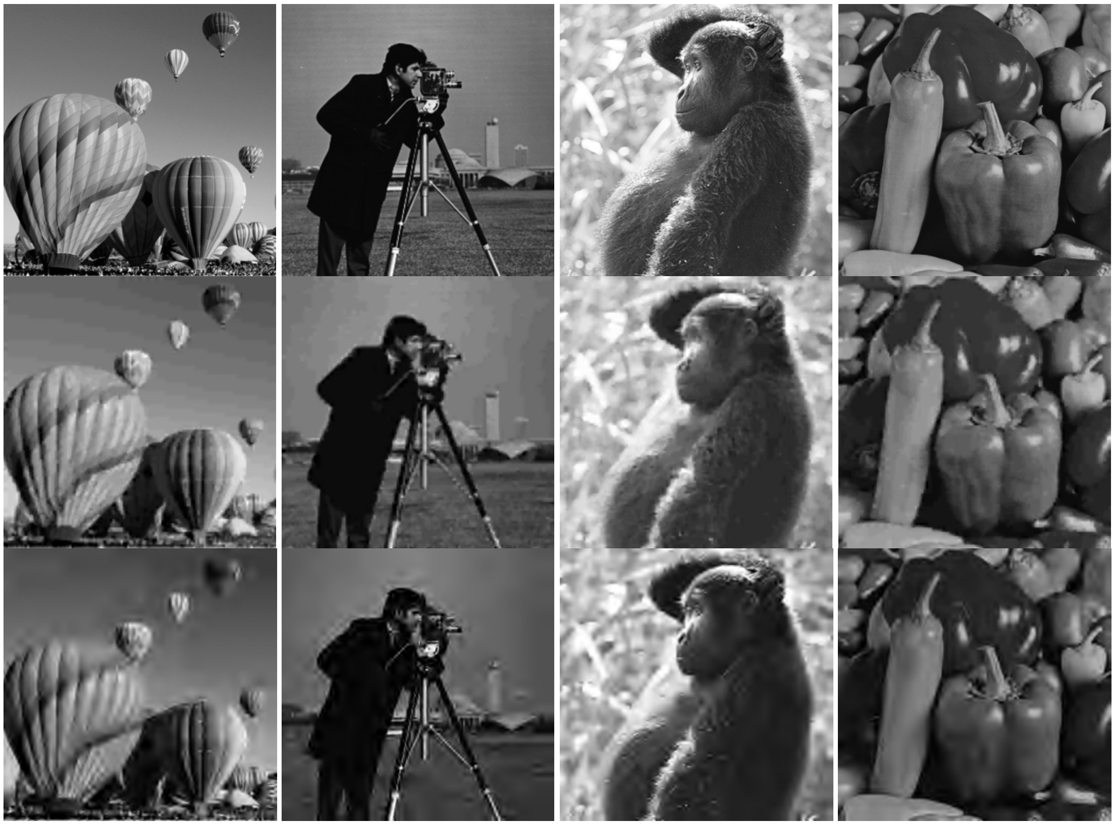


Fig 6.4 Images (original top row) used in a subjective comparison between the proposed method (middle row) and JPEG2000 (bottom row). Left to right are: Balloon, Cameraman, Chimpanzee, and Pepper.

Table 6.3 Percentage of viewers favouring the proposed method over JPEG2000 on 4 quality aspects.

Aspect Image	Edges	Interior Uniformity	Illumination	Information Preserving
Balloon	91	80	86	87
Cameraman	64	57	62	62
Chimpanzee	65	61	55	54
Pepper	74	57	57	54

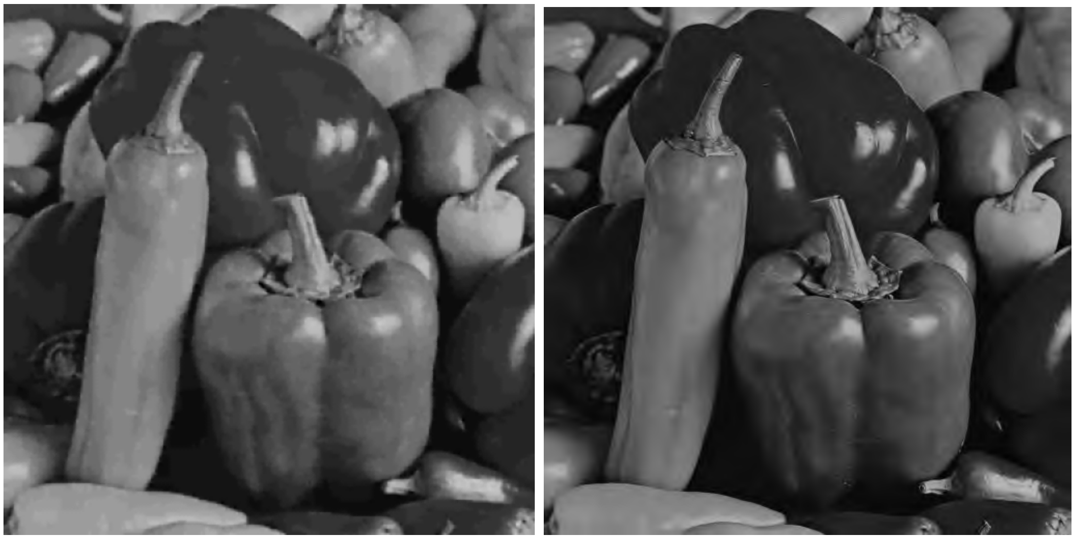


Fig 6.5 Reconstructed Images, (left) proposed at 35:1 and 31.1dB and (right) JPEG2000 at 33:1 and 22.7dB.

For objectivity, a comparison on image PEPPER using a different compression ratio is demonstrated in Fig 6.5. Again, JPEG2000 looks better for smooth regions but seems darker. However, for such low CR, JPEG2000 looks more pleasing while the proposed one seems noisier. The obvious reason is that only three parameters are used to describe each block. Hence, fine details will be lost and the available bit budget is spent on increasing the resolution of the three parameters that has negligible significance beyond a certain level.

Chapter 7

Postprocessing and Other Applications

The main objective is to find a subjectively correlated postprocessing scheme to enhance the reconstructed image (after decompression) by modifying Weber law [Ameer and Basir, 2008b]. The successful performance of the law is due to its correlation with the human visual system. Hence, it was also applied to image quality assessment. Interestingly, Weber law has some universality of implementing other image processing schemes, namely, edge detection [Ameer and Basir, 2008b]. The following sections give a detailed description of these three implementations.

7.1 Postprocessing Application

After applying the postprocessing scheme described in (4.9) or (4.10), a second enhancement stage is implemented using information from neighbouring blocks and incorporating a Weber-based postprocessing to obtain a better subjective quality of the reconstructed image. The proposed postprocessing scheme starts by finding Weber fraction between an object pixel I_o and a neighbouring background pixel I_b as given below:

$$W = \frac{I_o - I_b}{I_o} \tag{7.1}$$

Without loss of generality, the image is normalized to occupy the interval [0,1].

In fact, Weber fraction curve is symmetric as can be seen in Fig 2.3-1(a) of [Pratt, 2001]. This means that the eye response to dark and bright intensities is almost the same. This observation motivates the author to modify (7.1) to

$$W = \frac{I_o - I_b}{\max\{I_o, 1 - I_o\}} \tag{7.2}$$

W is then normalized to unity and thresholded. Equation (7.2) means that changes near 0.5 intensity have twice the effect of similar changes near 0 and 1 intensities on human perception. This comes in agreement with our eyes since we become more and more insensitive to changes as we approach white or black.

Before applying the previously described scheme, the image is divided into nonoverlapping blocks exactly as was done during compression. Pixels in the current block are grouped to form lines having slope angles of 0, ±π/4, and π/2, e.g., lines connecting (α, β) and (γ, δ) as shown in Fig 7.1.

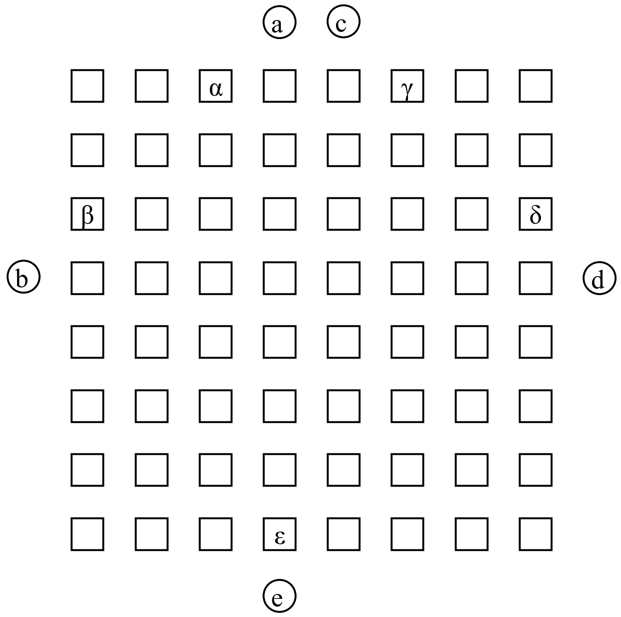


Fig 7.1 A sample demonstration for implementing inter-block enhancement

The mean of each line is found and subtracted from the two neighbouring points residing on the boundary of adjacent blocks. Mathematically, Equation (7.2) is modified to, for the case of the line connecting (α, β),

$$W_{\alpha,\beta} = \frac{\max\left\{\left|mean(\overline{\alpha\beta}) - a\right|,\ \left|mean(\overline{\alpha\beta}) - b\right|\right\}}{\max\left\{mean(\overline{\alpha\beta}),\ 1 - mean(\overline{\alpha\beta})\right\}} \tag{7.3}$$

If $\boldsymbol{W_{\alpha\beta}}$ is greater than 0.05, no postprocessing is performed. Otherwise, a line interpolating a and b is used to replace the segment αβ. A general description of the second postprocessing stage is presented in Fig 7.2. It should be noted that the lines described in Fig 7.2 are not of equal length. The factor 0.05 (approximating 1/16) was experimentally chosen. In addition, for horizontal and vertical lines, the postprocessing is implemented when prediction is not performed from the top and left neighbouring blocks respectively.

For each line, described in Fig 7.2, the average is obtained. However, since the reconstructed block values came from a plane model, each pixel differs from its two adjacent pixels by the same amount. The previous statement is essential to reduce the computation required for averaging to a single addition and one right shifting performed on the two ends of each line. The line is now represented by the average value obtained. Two pixels are then selected that are adjacent to the line end points, one for each end, from the neighbouring blocks.

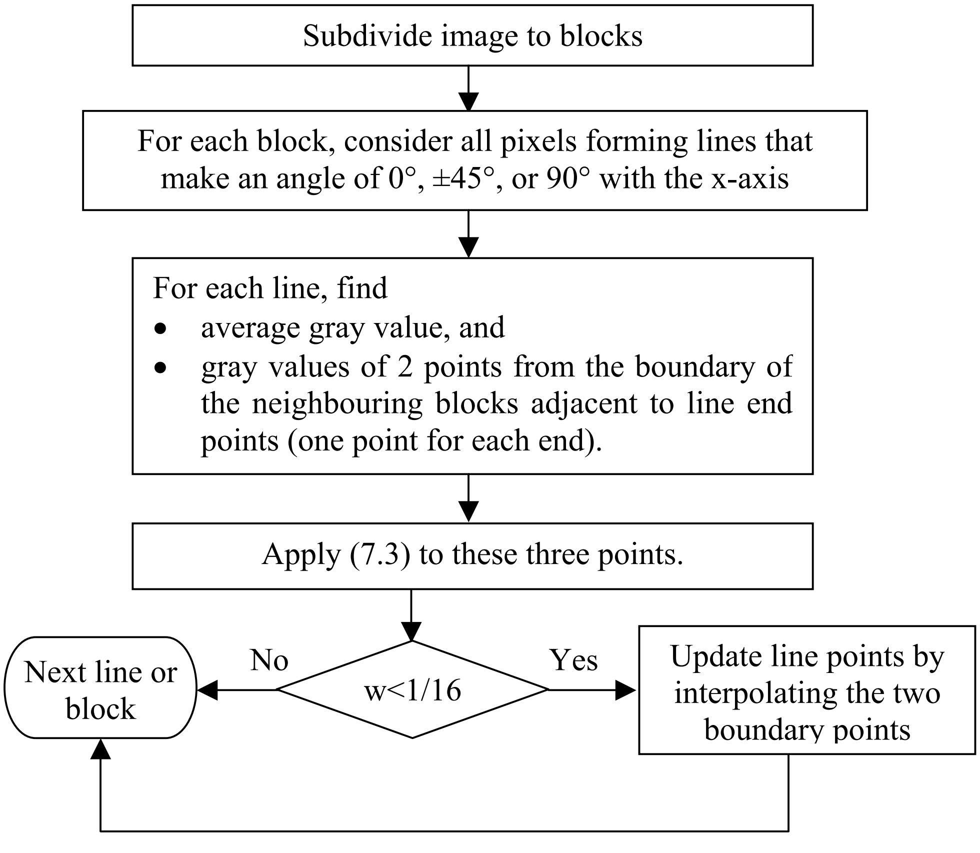

Subdivide image to blocks
For each block, consider all pixels forming lines that make an angle of 0°, ±45°, or 90° with the x-axis
For each line, find
• average gray value, and
• gray values of 2 points from the boundary of the neighbouring blocks adjacent to line end points (one point for each end).
Apply (7.3) to these three points.
w<1/16
No
Yes
Next line or block
Update line points by interpolating the two boundary points

Fig 7.2 Flow diagram of the proposed Weber-based postprocessing technique.

The three values, the average and the value of the two adjacent pixels, are plugged into (7.3) and the obtained parameter is compared to a threshold (1/16). If the threshold is exceeded, indicating the presence of edge, no enhancement is performed. Otherwise, the region is almost uniform and the postprocessing is performed. If the postprocessing is performed, the values of the two adjacent pixels of the neighbouring blocks are considered as belonging to a line of the same slope as the originally selected. Interpolation is then performed using these two points to find the new values of the in-between pixels.

7.1.1 Experimental Results

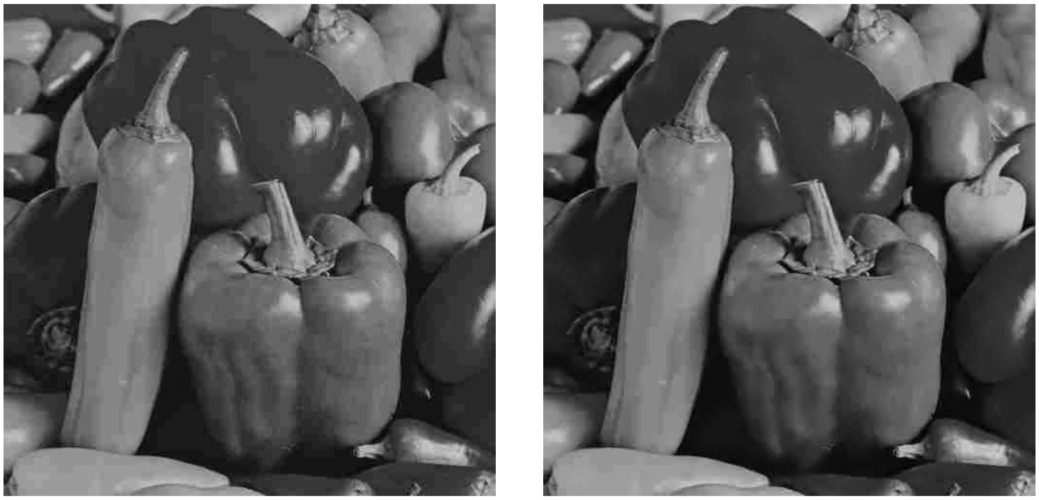


Fig 7.3 Reconstructed JPEG images (left) before postprocessing PSNR=32.2dB and (right) after postprocessing PSNR=32.4dB

Fig 7.3 illustrates the application of the proposed postprocessing scheme as in (7.3) on a JPEG version of the standard image PEPPER. Although the difference in PSNR is marginal ~0.2dB, the subjective improvement is remarkable especially on the large peppers. The same postprocessing is applied on the basic plane model of chapter 4. Again, as demonstrated in Fig 7.4, the difference in PSNR is < 0.1dB.

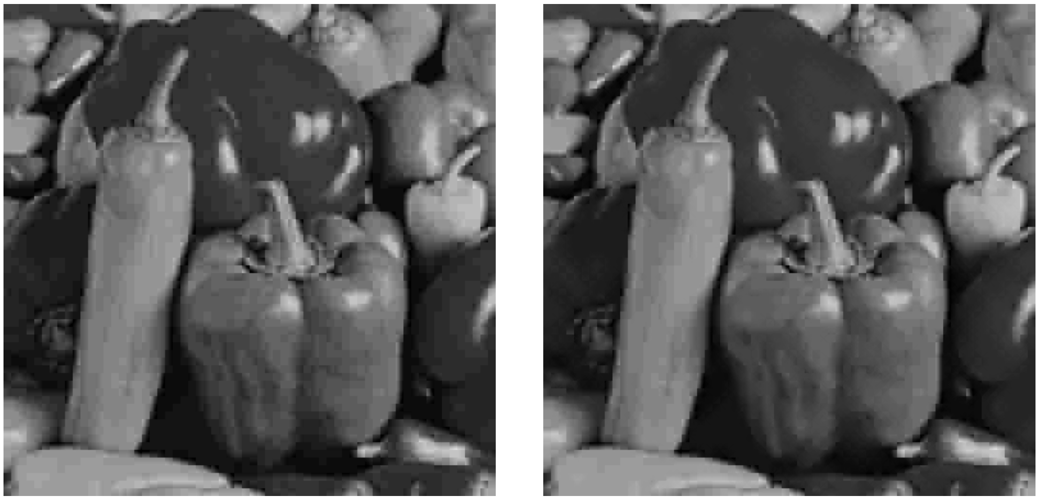


Fig 7.4 Reconstructed plane fitting images (left) before postprocessing PSNR=28.15dB and (right) after postprocessing PSNR=28.24dB

7.2 Quality Assessment Application

This section introduces three simple image quality measures based on Weber law. After normalizing the original and distorted images, the absolute error between them is found. The final index is a pixel-by-pixel weighted sum of errors. The maximum between the pixel value in the original image and its corresponding value in the negative image is the inverse of the weighting factor.

No claim is made concerning the superiority of the proposed scheme over any of the mentioned schemes below. However, when simplicity and good performance are both required; the proposed scheme could be of preference. Results indicate a better correlation with human perception compared to conventional mean square error MSE. Despite the simplicity of the proposed model, it compares well with other complex schemes in the literature and its complementary outcome is promising in improving those schemes especially with those having a data dependent parameter in the denominator.

7.2.1 The Proposed Scheme

Weber fraction can be easily modified to act as a QA scheme for images. The original image is considered as the background of the degraded image. The law is modified to overcome the biasedness toward dark intensities by an intuitive consideration of the experimental contrast sensitivity curve.

The proposed scheme starts by finding Weber fraction between all corresponding pixels in the original and degraded images as given below

$$W_i = \frac{I_i - \tilde{I}_i}{I_i} \tag{7.4}$$

where I is the original image, and $\tilde{I}$ is the degraded image. The images are normalized to occupy the interval [0,1].

In fact, the Weber fraction (contrast sensitivity) curve is symmetric, as can be seen in Fig 2.3-1(a) [Pratt, 2001]. This means that the eye response to differences in dark and bright intensities is almost the same. This observation motivates the author to modify (7.4) to obtain a Weber-based error (WE), given by

$$WE_i = \frac{I_i - \tilde{I}_i}{\max(I_i, 1 - I_i)} \tag{7.5}$$

Equation (7.5) means that changes near 0.5 intensity have twice the effect of similar changes near 0 or 1 intensity on human perception. This comes in agreement with our perception since we are more and more sensitive to changes occurring as we move away from either black or white.

The proposed Weber-based mean absolute error (WMAE) can be formulated as

$$WMAE = \frac{1}{N}\sum_i \frac{\left|I_i - \tilde{I}_i\right|}{\max(I_i, 1 - I_i)} \tag{7.6}$$

where N is the number of pixels in the image. The range of WMAE is from 0 to 1. A perfect match between the original and a degraded image results in WMAE=0. A decibel measure WPSNR (Weber-based peak signal to noise ratio) can be defined as,

$$WPSNR = 20\log_{10} \frac{N}{\sum_i \frac{\left|I_i - \tilde{I}_i\right|}{\max(I_i, 1 - I_i)}} \tag{7.7}$$

Equation (7.7) can be simplified to obtain,

$$WSNR = 20\log_{10} \frac{\sum_i \max(I_i, 1 - I_i)}{\sum_i \left|I_i - \tilde{I}_i\right|} \tag{7.8}$$

The three proposed schemes, Equations (7.6–7.8), are simple and can easily be applied in a rate distortion optimization.

7.2.2 Experimental Results

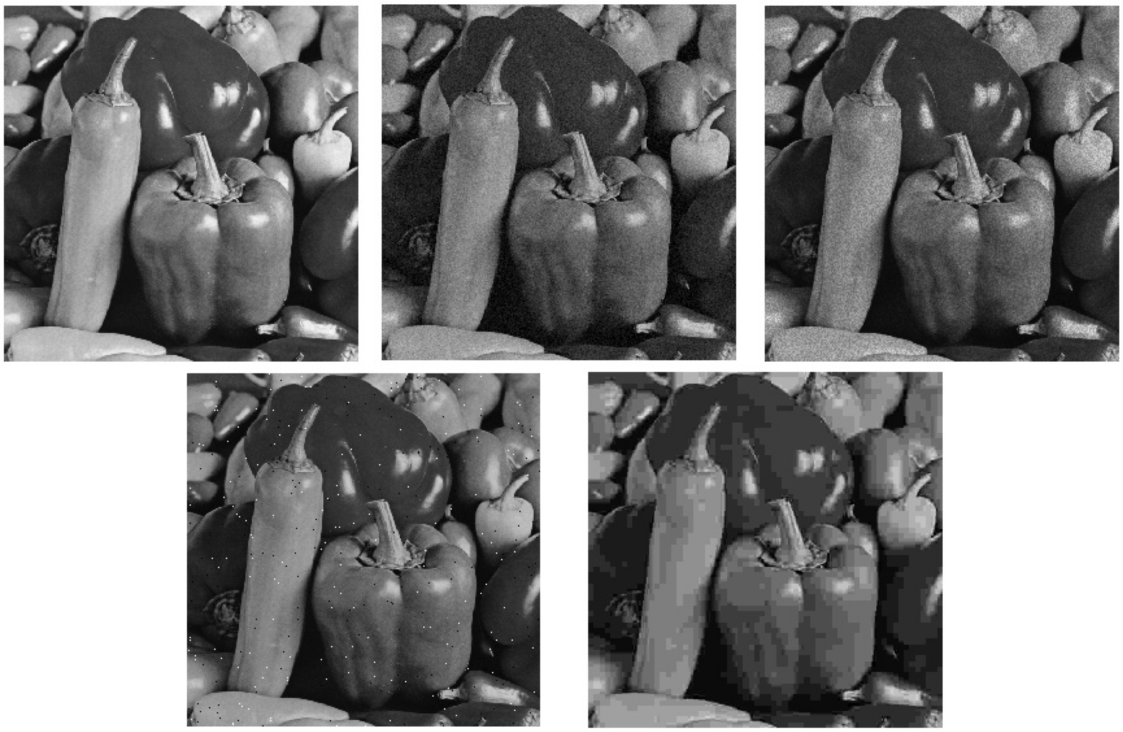


Fig 7.5 (left to right – top to bottom) Original, gaussian noisy, speckle noisy, salt and pepper (impulsive) noisy, and JPEG images used in QA testing. All degraded images have MSE=86.5.

Fig 7.5 shows the original image PEPPER (512x512) and four degraded images. Degradation comes in the form of gaussian noise, speckle noise, salt and pepper (impulse) noise, and JPEG compression. All degradations are chosen to result in an MSE of 86.5 before applying normalization to images.

Twenty-two viewers, most with image processing background, were asked to rank the 4 degraded images in descending quality. Table 7.1 shows viewers' statistics. Since the original image is always in rank 1 it is not shown in the table. Ranking is performed in this study due to a previous study [Ridder 1996] indicating that comparative testing is unbiased.

Table 7.1 Viewers' statistics for the **G**aussian noisy, **I**mpulse noisy, **S**peckle noisy, and **J**PEG images.

Images	**G**	**I**	**J**	**S**
Average Ranking	2.73	3.91	4.77	2.59
Count having rank 2	9	3	0	10
Count having rank 3	10	1	0	11
Count having rank 4	3	13	5	1
Count having rank 5	0	5	17	0

Table 7.2 QA for the **G**aussian noisy, **I**mpulse noisy, **S**peckle noisy, and **J**PEG images.

Images	**G**	**I**	**J**	**S**
WMAE	0.073	0.118	0.113	0.077
WSNR	23.19	19.17	19.51	22.84
WPSNR	22.80	18.57	18.98	22.24
PSNR	24.33	68.59	68.93	71.25
UIQI	50.98	82.05	37.66	60.63

The viewers were not told what type of degradation is applied to each image. The five images were seen together on the same screen. Equations (7.6–7.8) together with the universal image quality index (UIQI) of [Wang and Bovic, 2002] (using their own MatLab code) are calculated for the four images as shown in Table 7.2. For a fair comparison, the value of PSNR is calculated after performing image normalization.

The following two observations can be extracted from Table 7.1: 1) gaussian and speckle noisy images are of marginal difference and are both better than the rest, and 2) viewers are in favour of salt and pepper noise compared to JPEG compression. However, Table 7.2 shows that the proposed measures are in accordance with the first observation but not with the second. Interestingly, UIQI follows a complementary path where impulsive noisy image is second to none but the original image. As expected, PSNR is not in accordance with both observations.

7.2.3 Summary and Discussion

Simple image QA schemes were proposed based on weighting the mean absolute error according to Weber law. A rough agreement is noticed between the outcome of the proposed schemes and a subjective testing of 22 viewers.

It is observed that a complementary behaviour exists between the proposed schemes and that proposed by [Wang and Bovic, 2002]. Future work should concentrate on searching for a suitable fusing criterion to reach a better agreement with HVS. Further investigations are also required to relax the requirement for normalization. More experiments are needed on additional types of degradations to validate the reported results.

7.3 Edge Detection Application

This section introduces the design of an edge detection scheme the performance of which is pertinent to the properties of human visual system. The proposed scheme implements Weber fraction (also called contrast sensitivity) to detect edges in grey level images.

The expression $\Delta \boldsymbol{I}/\boldsymbol{I}$ is found between the current pixel and all its 8-neighbours and the maximum value is selected. To avoid biasing toward low intensity values and to preserve the symmetry of the sensitivity curve, $\boldsymbol{I}$ is replaced by the maximum of the actual pixel value and the corresponding value in the negative image. A thresholding procedure is then employed to eliminate weak edges. Experimental results indicate a superior capability of the proposed scheme to detect edges of objects that are close in intensity to their background. Some comparisons with Sobel operators are also demonstrated.

7.3.1 Introduction

Edge detection plays an important role in many computer vision applications. Several operators were designed in an ad hoc fashion to approximate the derivative for digitized images. The Sobel operator is an example of such commonly used difference operators. A good review of these operators can be found in [Ziou and Tabbone 1998].

Canny [Canny 1986] was the first to suggest an optimization criterion for finding edges using variational analysis. Although it opened a new trend in edge detection [Basu 2002], no considerations were given to the important player: human eye.

Motivated by proven facts on human visual system, many authors, e.g. [Pinoli 1997] and the references therein, introduced new definitions using logarithmic image processing. The most reliable theory, in my opinion, behind the success of logarithmic image processing is Weber fraction. The law [Pratt 2001] simply tells us that the ratio $\Delta \boldsymbol{I}/\boldsymbol{I}$ (keeping in mind that this ratio is the approximate derivative of the log function) governs the sensitivity of the eye such that no distinction between an object and its background can be made if this ratio is less than 0.02. This value is higher for darker and brighter intensity levels. $\Delta \boldsymbol{I}=(\boldsymbol{I}–\boldsymbol{I_0})$ is the intensity difference between the object and its background and $\boldsymbol{I}$ is the background (or foreground) intensity.

As a modification to overcome speckle noise in SAR images, the ratio between the intensity of neighbouring pixels (or regions inside a window) was used [Park *et al* 1995], [Bai and He 2003], and [Kang *et al* 2006]. However, these schemes are biased toward dark intensities due to division operation involved.

Another trend in modifying Weber fraction, though not explicitly stated, tackling the denominator rather than the numerator, was first proposed by [Yu 1994]. The same values in the numerator are used in the denominator but added not subtracted, in other words, $(\boldsymbol{I} – \boldsymbol{I_0}) / (\boldsymbol{I} + \boldsymbol{I_0})$ was proposed by [Yu 1994]. On the other hand, [Beghdadi *et al* 1999] proposed to use the average of a bank of Gabor filters to replace the denominator. In both schemes, the issue of biasedness is partially resolved.

In this work a new implementation of Weber fraction in edge detection to overcome the biasedness toward dark intensities (similarly unfairness toward bright intensities) is proposed through an intuitive consideration of the experimental contrast sensitivity curve.

7.3.2 The Proposed Scheme

The first phase of the proposed edge detection algorithm is finding the value of Weber fraction between the current pixel and all its eight neighbours. The equation for the left neighbour is given below as an example

$$W_{i,j,0,1} = \frac{I(i,j) - I(i,j-1)}{I(i,j)} \tag{7.9}$$

where ***I*** is the original image. Without loss of generality, the image intensity is normalized to the interval [0,1] by dividing by 255. In a similar fashion to (7.1), we can modify (7.9) to

$$W_{i,j,0,1} = \frac{I(i,j) - I(i,j-1)}{\max \{I(i,j), 1 - I(i,j)\}} \tag{7.10}$$

Combining the fractions from all 8-neighbours we have

$$W_{i,j} = \max_{m,n} \frac{I(i,j) - I(i-m,j-n)}{\max \{I(i,j), 1 - I(i,j)\}} \tag{7.11}$$

W is then normalized to the interval [0,1] and thresholded. Any thresholding scheme can be used, however, the following simple scheme outperforms Otsu [Otsu 1979] for a wide range of images. The proposed thresholding scheme for edge detection purposes is simply to find the histogram of ***W*** first. The histogram is then smoothed with an average filter. The position where the mean (on the smoothed histogram) occurs is taken as the threshold value. Usually, there is more than one position and the mean value is taken.

7.3.3 Experimental Results

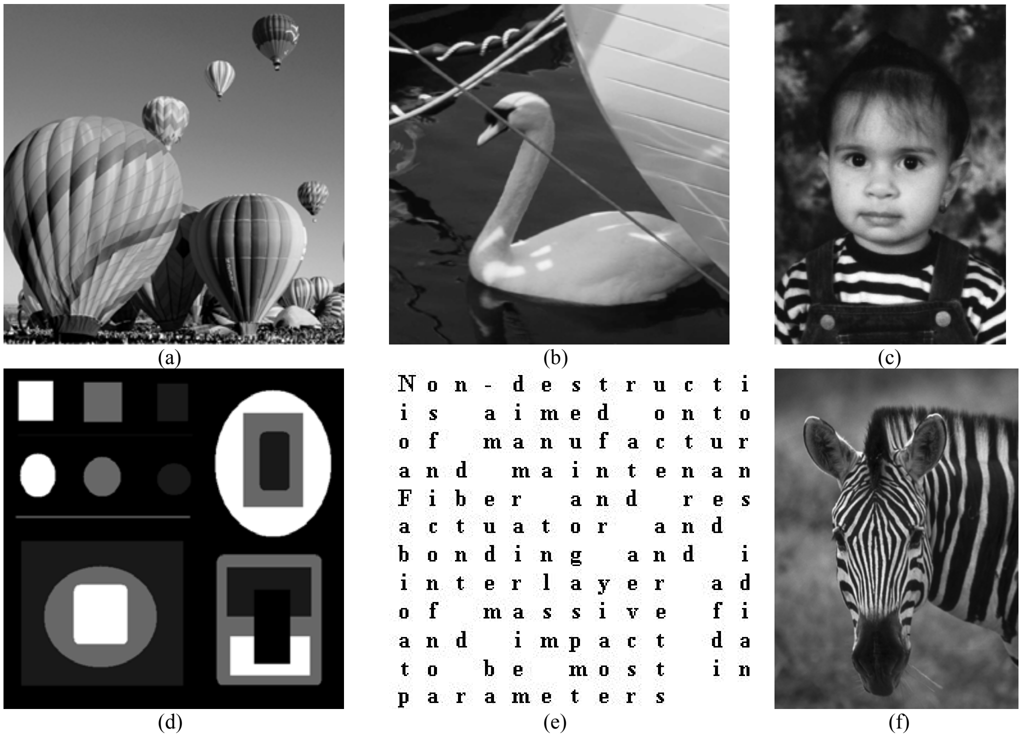
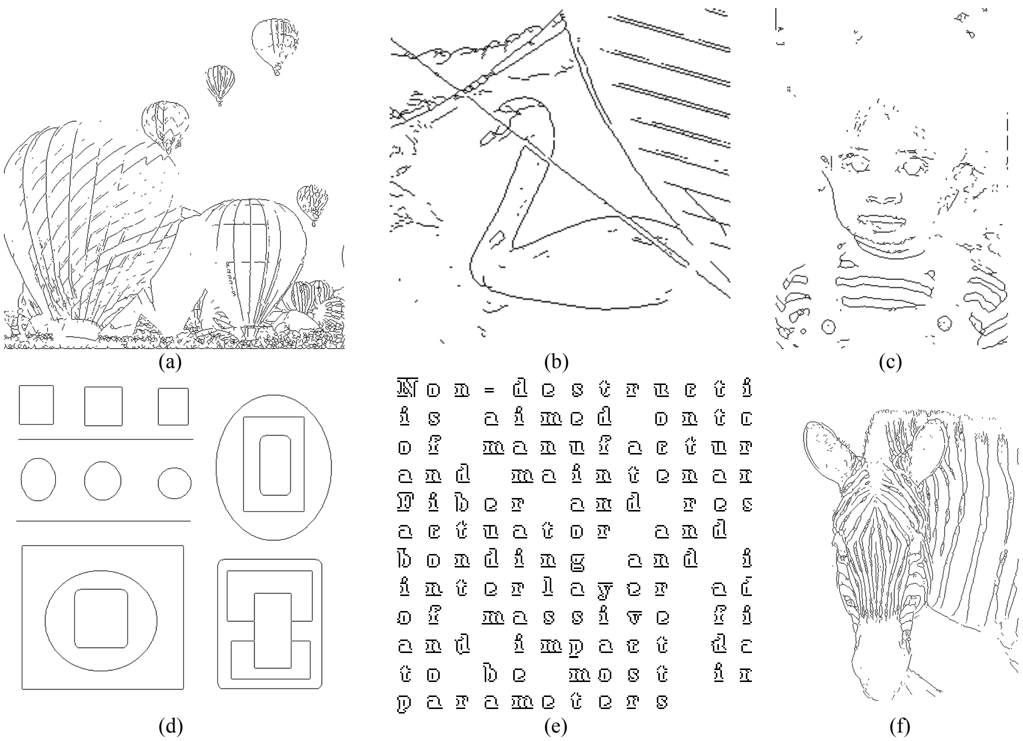


Fig 7.6 Test images: (a) Balloon, (b) Goose, (c) Huda, (d) Shapes, (e) Text, and (h) Zebra.

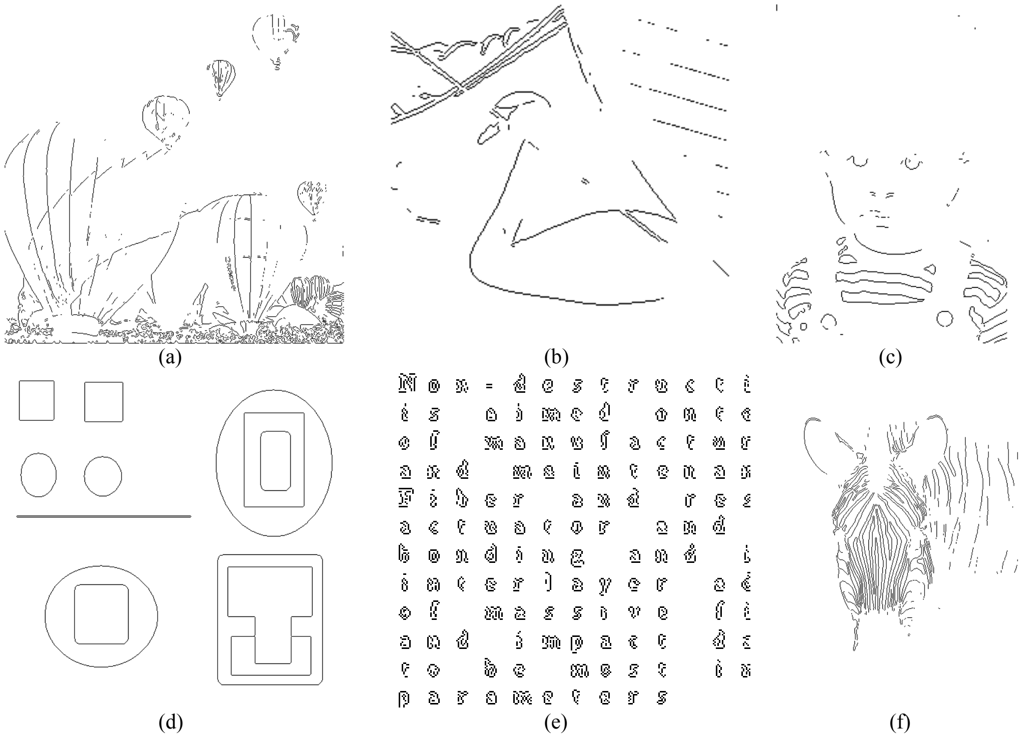
Fig 7.6 shows the set of images used to demonstrate the effectiveness of the proposed edge detection scheme using Weber fraction. Fig 7.7 shows the implementation of (7.11) on the set of images in Fig 7.6 after the thresholding scheme described in the last paragraph of the previous subsection. For comparison purposes, the same images are processed with the Sobel operator and presented in Fig 7.8. The images in Fig 7.7 and Fig 7.8 are after morphological thinning and cleaning operations.



(a) (b) (c)

(d) (e) (f)

Fig 7.7 Proposed edges for the images in Fig 7.6.


(a) (b) (c)

(d) (e) (f)

Fig 7.8 Sobel edges for the images in Fig 7.6.

The superiority of the proposed scheme is obvious through simple comparison of Fig 7.7 and Fig 7.8d where dark shapes and lines are considered as a background with Sobel that is not true by looking at Fig 7.6d. In general, the proposed scheme has a better capability than that of Sobel in closing object's boundaries as demonstrated by the examples in Fig 7.7 and Fig 7.8. Superiority is also clear for text images, obviously Fig 7.7e is more readable than Fig 7.8e.

7.3.4 Summary and Discussion

A simple edge detection scheme is proposed using Weber fraction. The model is designed to remove biasedness and unfairness towards dark and bright intensities respectively. More important, the author modifies the edge detection model to be more compatible with the actual curve of Weber fraction (contrast sensitivity).

The proposed operator shows better performance than that of Sobel for several images. However, the proposed model is in its early stages and there is still a room for improvement in various aspects. A better (especially local) thresholding scheme may be more appropriate to detect faint edges and exclude noisy ones. The scheme can also be applied on different resolution and then fused to obtain a better edge representation. In fact, finding the numerator of (7.5) on a filtered image (e.g., using a 5x5 average filter) and the denominator on another filtered version (e.g., using a 3x3 filter) can reduce the sensitivity to noise. This observation is helpful in designing the optimum filter in a similar fashion to the scheme of [Canny 1986].

Chapter 8

Conclusions and Future Work

Surface fitting schemes have been explored in this work for image compression purposes. The performance, using basic plane fitting model, is superior (both perceptually and in PSNR value) to that of JPEG at compression ratios >32:1. In the proposed scheme, each block is represented by three parameters, namely ***a***, ***b***, and ***c***. Compression is performed by sending codes representing the quantized values of these parameters. The quantization is uniform for ***c*** and nonuniform for ***a***, and ***b***. To reduce quantization error [Strobach 1991] and the number of bits allocated to the constant parameter ***c***, the block centre is chosen as the origin of the coordinate system. Simple quantization, in the form of a lookup table, and coding scheme (comma or Huffman coding) are used to reduce cost. The quantization table is designed off-line, using some set of images, through a direct search method compromising CR and PSNR. A further research is needed to study the effects on other image sets and whether different tables are needed for each set of images.

Plane fitting is a multiplication- and division-free block-based image compression implementation. The number of shifts can be drastically decreased at the decoder by adopting similar calculations to that of [Hasegawa and Yamasaki 2002]. This low computational cost makes the proposed algorithm suitable for real time applications. Embedded coding can be achieved by sending ***c*** on bit bases followed by ***a***(***b***) and ***b***(***a***). In fact, sending functions of ***a*** and ***b*** can slightly increase CR keeping PSNR almost unaffected. This observation needs further investigations together with a more perceptually correlated error measure to optimize the compression performance.

To enhance the reconstructed image, a simple 2-point 1D linear interpolation is proposed to reduce blocking artefacts. The obtained reduction compares well to that obtained using 10-point cubic fitting. In addition, the proposed interpolation scheme can be is modified to maintain the multiplication- and division-free characteristic. The proposed postprocessing scheme can increase PSNR of the reconstructed image by <1dB depending on the image under consideration.

Another contribution of this work is the incorporation of a perceptually related law; namely the Weber fraction, in postprocessing of reconstructed images. Its implementation is not restricted to the plane fitting scheme but can be applied to other schemes, like JPEG for example. The performance is rather subjective and further research is required to better formulate and understand this phenomenon. In addition, Weber law has some universality in image processing in the sense that it can be modified to produce two more schemes, namely, edge detection, and quality assessment. The Weber-based postprocessing scheme has marginal effect on PSNR value but remarkably improves the subjective quality. In this scheme, neighbouring blocks contribute to the enhancement of part of the current block depending on satisfying a condition that is mainly a revised version of the edge detection scheme using Weber law. It should be stressed that the amount of remarkable improvements gained from the two proposed postprocessing schemes indicates a possibility of strong correlation, and hence a possibility of a more compact form, between blocks. This correlation is worth to be further investigated.

It is observed that the profile of ***a***(***b***) in the plane model has a strong similarity with horizontal (vertical) edge profile. This interesting observation is important to find better edge and/or texture descriptions to improve visual quality and to a less extent compression ratio. In fact, an edge of any direction can be inferred (to some acceptable error) from the absolute ratio of ***a*** and ***b***. It should be emphasised that the previous statement is not about edge strength only but more importantly its orientation. However, further information is needed to find a good approximation of edge location. The main challenge in transforming such observation to a useful implementation is the appropriate function combining ***a*** and ***b*** to describe oblique edges. A more challenging aspect, though low in practice, is the possibility that more than one edge can occur in the same block. Similar argument also applies to nonlinear edges. These multiedges can pass undetected or erroneously detected as a single edge. Neural networks and/or fuzzy systems can be useful to overcome this problem.

The current implementation of the plane model is concerned with gray scale images. The extension to coloured images is straightforward. However, colour transformations should be avoided (or manipulated) to preserve the multiplication-free characteristic. Extension to video coding is also possible by implementing the plane model as the intra-frame mode and motion compensation for inter-frame prediction. A comparison should be performed between different motion compensation schemes, especially the linear model, and the 3D extension of the proposed model.

As is clearly seen in the results of chapter 4, most of the allocated bits (in the basic plane model) are spent on coding the quantized value of ***c***. Hence, it may be useful to predict ***c*** from neighbouring blocks to increase CR. Unlike the prediction of the DC component in JPEG, prediction is performed with the help of the boundary pixels of the neighbouring blocks. To the author's knowledge, this form of prediction has not been reported. The predicted value is also dependent on the values of ***a*** and ***b*** of the current block. The constrained optimization problem is solved for ***a*** and ***b***. Compression ratio is further increased by eliminating the need for sending the index bit, indicating the position of the neighbouring block to predict from, if the predicted value of ***c*** is the "same" for all possibilities. The condition "same" can be relaxed to further increase CR, however, quality may degrade noticeably. Interestingly, the plane fitting scheme with inter-block prediction favours (in some cases) JPEG2000 at high compression ratios despite the fact that JPEG2000 is not a block-based compression scheme. The quality is slightly better both in terms of PSNR and subjectively through the statistics of 22 viewers. The proposed plane model, in its two forms, is a highly parallel structure. This aspect can be of great importance when a fast implementation is required for low bit rate moderate quality applications like sending head and shoulder images using mobile phones for example.

Unfortunately, the plane model cannot describe fine structures in the image and more parameters are needed and hence, higher order polynomials are investigated.

These higher order models can be useful, through some aggregation process, in improving the plane model performance for low compression rate applications. However, these high order schemes are restricted (in the present study) to three parameters only to maintain the improved compression performance obtained with the plane model. Among all 2D nonlinear polynomials presented in this work, the *xy* model is the only scheme that has comparable (except for complexity) performance to that of the plane model. Combining this model with the plane model can be fruitful if the fusion is dynamic with as small header bits as possible.

Going beyond second order, keeping in mind the 3-parameter restriction, is not mathematically tractable and the performance is not expected to substantially improve as can be deduced from sections 5.1 – 5.4 of this work. Therefore, a linear mapping scheme is implemented to get some insight of an approximate asymptotic behaviour. Interestingly, the results of this mapping scheme (section 5.5) are comparable to that of the plane model. This evidence reveals the power inherent in the plane model despite its simplicity. Nevertheless, the proposed linear mapping scheme can provide a new framework for image modeling.

Though polynomial fitting is promising, however, it is constrained by the block size. In general, performance can be increased, at least theoretically, by increasing the polynomial order and to a less extent the block size. By relaxing these two constraints, i.e. using polynomials and dividing the image into blocks, it may be possible to overcome the current barrier in the amount of compression attained. Hence, an investigation of nonpolynomial models is required. The objective is to increase compression ratio through a better description of the objects or subjects in the image and relaxing the constraint of dividing the image into blocks. In other words, the concern is rather on image context through the efficient description of the composing objects. A simple criterion is probably to view an image as a 3D geographical map consisting of hills and valleys. It should be kept in mind, however, that such modelling is of high computational complexity and a deeper insight is required to overcome this drawback. Another suggestion is design a new formulation that can overcome the disadvantage of having a separate optimization and coding schemes to describe a region and its boundary. Stated differently, the formulation should somehow combine a region and its contour, and perhaps portions or nearby regions, in a single coding paradigm without the need for a segmentation process.

Region fitting approach, as the least successful proposal, needs further and deeper insight. The proposed approach is a new research trend in combining a region and its contour in a single coding paradigm. The MSE optimization, in general, tries to fit the largest possible number of points to any specific region. This will create "virtual" regions that require additional bits and computational time. A simple optimization scheme using Hooke and Jeeve's (HJ) scheme is implemented to validate the approach. The results are encouraging in terms of PSNR; however, the computational burden is far from being practical. The performance is inferior to state of the art image compression; however, the potential in implementing region-based scheme with a single optimization pass is promising. The reason for these drawbacks may be related to the fact that iterative (not direct) solutions were found. Highly complex mathematical manipulations of the problem may be needed to benefit from the speed of Newton and quasi-Newton implementations. In addition, Hessian matrix ill conditioning and trapping in a local minimum need some compensation. Finding more than one function at a time may be part of the solution by incorporating steepest descent methods. Taking benefit of ***a*** and ***b*** in the plane model together with some morphological operations to seed the region fitting scheme is of marginal effect in reducing computational complexity.

References

H. Abbas, 2007. Classified image compression using optimally structured auto-association networks. *IET Image Process.* 1 (2), pp. 189–196.

A. Abbas and T. Tran, 2006. Multiplierless design of biorthogonal dual-tree complex wavelet transform using lifting scheme. *Int. Conf. on Image Processing ICIP*, pp 1605–1608.

T. Acharya and P. Tsai, 2005, JPEG2000 standard for image compression: concepts, algorithms and VLSI architectures. *John Wiley and Sons Inc*.

S. Agaian, T. Baran, and K. Panetta, 2003. The application of logical transforms to lossless image compression using Boolean minimization. *Proceedings, GSPx and ISPC*, pp 13 – 31.

V. Algazi, N. Avadhanam, and R. Estes, 1998. Quality measurement and use of pre-processing in image compression. *Signal Processing 70*, pp 215 – 1229.

A. Alkholidia, A. Alfaloua, and H. Hamam, 2007. A new approach for optical colored image compression using the JPEG standards. *Signal Processing 87*, pp 569–583.

H. Al-Otum, 2003. Qualitative and quantitative image quality assessment of vector quantization, JPEG, and JPEG2000 compressed images. *Journal of Electronic Imaging* 12(3), pp 511 – 521.

T. Amarunnishad, V. Govindan, and A. Mathew, 2008. Improving BTC image compression using a fuzzy complement edge operator. *Signal Processing 88*, pp 2989–2997.

S. Ameer and O. Basir, 2006. A simple three–parameter surface fitting scheme for image compression. *Int. Conf. on Computer Vision Theory and Applications VISAPP, Setúbal, Portugal*, pp 101–106.

S. Ameer and O. Basir, 2008a. Image Compression through Optimized Linear Mapping and Parametrically Generated Features. *International Symposium on Communications, Control and Signal Processing ISCCSP, St. Julians, Malta*, pp 867 – 870.

S. Ameer and O. Basir, 2008b. Modifying Weber fraction law to postprocessing and edge detection applications. *International Symposium on Communications, Control and Signal Processing ISCCSP, St. Julians, Malta*, pp 871 – 875.

S. Ameer and O. Basir, 2009. Image Compression Using Plane Fitting with Inter-Block Prediction. Image and Vision Computing 27, pp 385 – 390.

S. Ameer and B. Shahrrava, 2005. Multiplier-free NLMS for adaptive IIR filtering. *IEEE Canadian Conf. on Electrical and Computer Engineering, Saskatoon Inn, Sk, Canada*, pp 1237–1240.

I. Aravind, C. Chaitanya, M. Guruprasad, S. Partha, S. Sudhaker, 2002. Implementation of image segmentation and reconstruction using genetic algorithms. *IEEE Int. Conf. on Industrial Technology*, pp 970 – 975.

J. Augustine, W. Lynch, Y. Wang, and A. Al-Khalili, 1999. Lossy compression of images using logic minimization. *Proc. of 12th International Conference on VLSI Design, VLSI'99,* Goa, India, pp 7 – 10.

I. Avcibas, B. Sankur, and K. Sayood, 2002. Statistical evaluation of image quality measures. *Journal of Electronic Imaging* 11(2), pp 206 – 223.

H. Aydinoglu and M. Hayes, 1996. Image coding with polynomial transforms. *13th Asilomar Conf. on Signals, Systems, and Computers, Vol. 1*, pp 520 – 524.

Z. Bai and P. He, 2003. An improved ratio edge detector for target detection in SAR images. *IEEE Int. Conf. Neural Networks and Signal Processing, Nanjing, China*, pp 982–985.

R. Baseri and J. Modestino, 1994. Region-based coding of images using a spline model. *Proc. IEEE Int. Conf. Image Processing, Vol. 3*, pp 866 – 870.

M. Basu, 2002. Gaussian-Based Edge-Detection Methods—A Survey. *IEEE Trans. Systems, Man, and Cybernetics–Part C: Applications and Reviews, Vol. 32, No. 3*, pp 252–260.

A. Beghdadi, K. Belkacem-Boussaid and A. Boudraa, 1999. Low-level vision treatments inspired from human visual system. 5th *Int. Symposium on Signal Processing and its Applications, ISSPA, Australia,* pp 313–316.

A. Berg and W. Mikhael, 1994. A survey of techniques for lossless compression of signals. *Proc. 37th Midwest Symposium on Circuits and Systems, Vol. 2*, pp 943 – 946.

V. Bhaskaran and K. Konstantinides, 1997. Image and video compression standards – algorithms and architectures. 2nd ed. Kluwer Academic.

M. Biggar, O. Morris, A. Constantinides, 1988. Segmented-image coding: performance comparison with the discrete cosine transform. *IEE Proc., Vol. 135, Pt. F, No. 2*, pp 121 – 132.

S. Biswas, 2003. Segmentation based compression for gray level images. *Pattern Recognition 36*, pp 1501 – 1517.

J. Bosworth and S. Acton, 2000. Segmentation-based image coding by morphological local monotonicity. *34th Asilomar Conf. on Signals, Systems and Computers*, pp 65 – 69.

A. Bouzerdoum, A. Havstad, and A. Beghdadi, 2004. Image quality assessment using a neural network approach. *Proc. 4th IEEE Int. Symposium on Signal Processing and Information Technology*, pp 330 – 333.

J. Brankov, I. El-Naqa, Y. Yang, and M. Wernick, 2003. Learning a nonlinear channelized observer for image quality assessment. *IEEE Nuclear Science Symposium Conf. Record, Vol.4,* pp 2526 – 2529.

V. Bruni and D. Vitulano, 2007. Combined image compression and denoising using wavelets. *Signal Processing: Image Communication 22*, pp 86–101.

C. Cabrelli and U. Molter, 1990. Automatic representation of binary images. *IEEE Trans. PAMI, Vol. 12, No. 12*, pp 1190 – 1196.

J. Canny, 1986. A computational approach to edge detection. *IEEE PAMI 8(6)*, pp 679–698.

M. Cermelli, F. Lavagetto, and M. Pampolini, 1994. A fast algorithm for region-oriented texture coding. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Vol. 5*, pp 285 – 288.

R. Chan and M. Lee, 2006. Multiplierless Fast DCT Algorithms with Minimal Approximation Errors. *18th Int. Conf. on Pattern Recognition (ICPR'06)*, pp 921 – 925.

C. Chang and B. Girod, 2007. Direction-adaptive discrete wavelet transform for image compression. *IEEE Trans. IP, Vol. 16, No. 5*, pp 1289 - 1302.

Y. Chen, H. Ten, and W. Hsu, 1994. Compression of colour image via the technique of surface fitting. *Graphical Models and Image Processing, Vol. 13, No. 10*, pp 272 – 279.

H. Chen and P. Varshney, 2007. A human perception inspired quality metric for image fusion based on regional information. *Information Fusion 8*, pp 193–207.

S. Choy, Y. Chan, and W. Siu, 1996. An improved quantitative measure of image restoration quality. *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing ICASSP, Vol. 3*, pp 1613 – 1616.

S. Costa and S. Fiori, 2001. Image compression using principle component neural networks. *Image and Video Computing* 19, pp 649–668.

S. Daly, 1992. The visible differences predictor: an algorithm for the assessment of image fidelity. *SPIE Vol. 1666 Human Vision, Visual Processing, and Digital Display III*, pp 2 – 15.

B. Dasarathy, 1995. Image data compression: block truncation coding. *IEEE Computer Society Press*.

E. Delp and O. Mitchell, 1979. Image compression using block truncation. *IEEE Trans. Comm., Vol. Com-27, No. 9*, pp 1335 – 1342.

L. Demaret, N. Dyn, and A. Iske, 2006. Image compression by linear splines over adaptive triangulations. *Signal Processing 86*, pp 1604–1616.

F. DeNatale, G. Desoli, D. Giusto, and G. Vernazza, 1995. Polynomial approximation and vector quantization: a region-based integration. *IEEE Trans Comm., Vol. 43, No. 2/3/4*, pp 198 – 206.

B. Dhara and B. Chanda, 2004. Block truncation coding using pattern fitting. *Pattern Recognition 37*, pp 2131 – 2139.

B. Dhara and B. Chanda, 2007. Color image compression based on block truncation coding using pattern fitting principle. *Pattern Recognition 40*, pp 2408 – 2417.

F. Di Martino and S. Sessa, 2007. Compression and decompression of images with discrete fuzzy transforms. *Information Sciences 177*, pp 2349–2362.

A. Di Nola and C. Russo, 2007. Łukasiewicz transform and its application to compression and reconstruction of digital images. *Information Sciences 177*, pp 1481–1498.

R. Distasi, M. Nappi, and D. Riccio, 2006. A Range/Domain Approximation Error-Based Approach for Fractal Image Compression. *IEEE Trans. IP, Vol. 15, No. 1*, pp 89 - 97.

Z. Dokur, 2008. A unified framework for image compression and segmentation by using an incremental neural network. *Expert Systems with Applications 34*, pp 611–619.

M. Eckert and A. Bradley, 1998. Perceptual quality metrics applied to still image compression. *Signal Processing 70*, pp 177 – 200.

M. Eden, M. Unser, and R. Leonardi, 1986. Polynomial representation of pictures. *Signal Processing 10*, pp 385–393.

O. Egger, P. Fleury, T. Ebrahimi, and M. Kunt, 1999. High-performance compression of visual information—a tutorial review—Part I: still pictures. *Proc. IEEE, Vol. 87, No. 6*, pp 976 – 1011.

O. Elbadawy, M. El-Sakka, and M. Kame1, 1998. An information theoretic image-quality measure. *IEEE Canadian Conf. on Electrical and Computer Engineering, Vol. 1*, pp 169 – 172.

A. Eskicioglu and P. Fisher, 1993. A survey of quality measures for gray scale image compression. *Space and Earth Science Data Compression Workshop*, Ed. J. Tilton, pp 49 – 61.

A. Eskicioglu and P. Fisher, 1995. Image quality measures and their performance. *IEEE Trans. Comm., Vol. 43, No. 12*, pp 2959 – 2965.

T. Eudu and A. Mayache, 1998. An evaluation of quality metrics for compressed images based on human visual sensitivity. *Proc. 4th Int. Conf. on Signal Processing ICSP, Vol. 1*, pp 779 – 783.

H. Feng, C. Chen, and F. Ye, 2007. Evolutionary fuzzy particle swarm optimization vector quantization learning scheme in image compression. *Expert Systems with Applications 32*, pp 213–222.

S. Fernandez, R. Estepar, C. Lopez and C. Westin, 2006. Image Quality Assessment based on Local Variance. *Proc. 28th IEEE EMBS Annual Int. Conf., New York City, USA*, pp 4815 – 4818.

P. Franti, 1998. Blockwise distortion measure for statistical and structural errors in digital images. In *Signal Processing: Image Communication 13*, pp 89–98.

D. Fuhrmann, J. Baro, and J. Cox, 1995. Experimental evaluation of psychophysical distortion metrics for JPEG-encoded images. *J. of Electronic Imaging* 4(4), pp 397 – 406.

B. Furht, 1995. A survey of multimedia compression techniques and standards Part I: JPEG Standard. *Real-time Imaging*, pp 49–76.

M. Gilge, 1990. Region-oriented transform coding (ROTC) of images. *Int. Conf. on Acoustics, Speech, and Signal Processing*, pp 2245 – 2248.

G. Ginesu, F. Massidda, and D. Giusto, 2006. A multi-factors approach for image quality assessment based on a human visual system model. *Signal Processing: Image Communication 21*, pp 316–333.

D. Giusto and M. Perra, 1997. Estimating blockness distortion for performance evaluation of picture coding algorithms. *IEEE Pacific Rim Conf. on Communications, Computers and Signal Processing, Vol. 1*, pp 318 – 321.

A. Habibi, 1971. Comparison of nth-order DPCM encoder with linear transformations and block quantization techniques. *IEEE Trans. Comm., Vol. 19, No. 6*, pp 948 – 956.

A. Habibi, 1977. Survey of adaptive image coding technique. *IEEE Trans. Comm., Vol. 25, No. 11*, pp 1275 – 1284.

M. Hasegawa and I. Yamasaki, 2002. Image data compression with nonuniform block segmentation and luminance approximation using bilinear curved surface patches. *Systems and Computers in Japan, Vol. 33, No. 10*, pp 31 – 40.

S. Hemami and R. Gray, 1997. Subband-coded image reconstruction for lossy packet networks. *IEEE Trans. IP, Vol. 6, No. 4*, pp 523 – 539.

R. Hooke and T. Jeeves, 1961. Direct search solution of numerical and statistical problems. *Journal of the ACM, Vol. 8*, pp. 212 – 229.

I. Hussain and T. Reed, 1994. Segmentation-based image compression with enhanced treatment of textured regions. *28th Asilomar Conf. on Signals, Systems and Computers*, pp 965 – 969.

E. Iyoda, T. Shibata, H. Nobuhara, W. Pedrycz, and K. Hirota, 2007. Image compression and reconstruction using pi_t-sigma neural networks. *Soft Comput 11*, pp 53–61.

D. Jackson, H. Ren, X. Wu, and K. Ricks, 2007. A hardware architecture for real-time image compression using a searchless fractal image coding method. *J Real-Time Image Proc 1*, pp 225–237.

A. Jain, 1981. Image data compression: a review. *Proc. IEEE, Vol. 69, No. 3*, pp 349 – 401.

J. Jain and A. Jain, 1981. Displacement measurement and its application in interframe image coding. *IEEE Trans. Comm., Vol. 29, No. 12*, pp 1799 – 1808.

J. Jiang, 1999. Image compression with neural networks – a survey. *Signal Processing: Image Communication 14*, pp 737 – 760.

X. Kang, C. Han, Y. Yang, and T. Tao, 2006. SAR image edge detection by ratio-based Harris method. *IEEE ICASSP* 2006, pp II-837–II-840.

M. Karczewics, J. Nieweglowski, and P. Haavisto, 1997. Video coding using motion compensation with polynomial motion vector fields. *Signal Processing: Image Communication 10*, pp 63 – 91.

A. Kaup and T. Aach, 1994. Segment-Oriented Coding of Textured Images Based on Successive Approximation. *Int. Symposium on Speech, Image Processing and Neural Networks, Hong Kong*, pp 197 – 200.

A. Kaup and T. Aach, 1998. Coding of segmented images using shape-independent basis functions. *IEEE Trans IP, Vol. 7, No. 7*, pp 937 – 947.

L. Kaur, R.Chauhan and S. Saxena, 2006. Adaptive compression of medical ultrasound images. *IEE Proc.-Vis. Image Signal Process., Vol. 153, No. 2*, pp 185 - 190.

T. Kieu and D. Nguyen, 2001. Surface fitting approach for reducing blocking artifacts in low bit-rate DCT decoded images. *Proc. IEEE Region 10 Int. Conf. on Electrical and Electronic Technology, Vol. 1*, pp 23 – 27.

H. Kim and J. Lee, 2002. Image coding by fitting RBF-surfaces to subimages. *Pattern Recognition Letters 23*, pp 1239–1251.

A. Kingston and F. Autrusseau, 2008. Lossless image compression via predictive coding of discrete Radon projections. *Signal Processing: Image Communication 23*, pp 313– 324.

A. Laha, B. Chanda, and N. Pal, 2008. Fast codebook searching in a SOM-based vector quantizer for image compression. *SIViP 2*, pp 39–49.

A. Laha, N. Pal, and B. Chanda, 2004. Design of vector quantizer for image compression using self-organizing feature map and surface fitting. *IEEE Trans. IP, Vol. 13, No. 10*, pp 1291 – 1303.

Y. Lai and C. Kuo, 2000. A Haar wavelet approach to compressed image quality measurement. *J. of Visual Communication and Image Representation 11*, pp 17 – 40.

Y. Lan, A. Chen, J. Lin, and K. Wen, 1998. Three-dimensional regression polynomial coding for video transmission. *The 9th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Vol.3,* pp 1071 – 1075.

J. Li, G. Chen, Z. Chi, and C. Lu, 2004. Image coding quality assessment using fuzzy integrals with a three-component image model. *IEEE Trans. Fuzzy Systems, Vol. 12, No. 1*, pp 99 – 106.

W. Li and Y. Zhang, 1995. Vector-based signal processing and quantization for image and video compression. *Proc. IEEE, Vol. 83, No. 2*, pp 317 – 335.

Y. Lim and K. Park, 1998. Image segmentation and approximation through surface type labelling and region merging. *Elect. Lett., Vol. 24, No. 22*, pp 1380 – 1381.

J. Lin and M. Smith, 2008. New perspectives and improvements on the symmetric extension filter bank for subband/wavelet image compression. *IEEE Trans. IP, Vol. 17, No. 2*, pp 177 - 189.

Y. Lin and P. Vaidyanathan, 1996. Theory and design of two-dimensional filter banks: a review. *Multidimensional Systems and Signal Processing 7*, pp 263 – 330.

Z. Liu and R. Laganiere, 2007. Phase congruence measurement for image similarity assessment. *Pattern Recognition Letters 28*, pp 166 – 172

T. Lu, Z. Le, and D. Yun, 2000. Piecewise Linear Image Coding Using Surface Triangulation and Geometric Compression. *Proc. Data Compression Conference*, pp 410 – 419.

L. Ma, and K. Khorasani, 2002. Application of adaptive constructive neural networks to image compression. *IEEE Tran. Neural Networks* 13(5), pp 1112–1126.

H. Marmolin, 1986. Subjective MSE measures. *IEEE Trans. Sys, Man, and Cyber, Vol. SMC-16, No. 3*, pp 486–489.

J. Max, 1960. Quantizing for minimum distortion. *IRE Trans. IT, Vol. 6, No. 2*, pp 7 – 12.

L. Meesters, W. IJsselsteijn, and P. Seuntiëns, 2004 . A survey of perceptual evaluations and requirements of three-dimensional TV. *IEEE Trans. CAS-VT, Vol. 14, No. 3*, pp 381 – 391.

M. Mittal, V. Singh, and R. Krishnan, 1999. A new criteria for evaluation of compression technique. *5th Int. Symposium on Signal Processing and its Applications, ISSPA '99, Brisbane, Australia*, pp 297 – 300.

M. Miyahara, K. Kotani, and V. Algazi, 1998. Objective picture quality scale (PQS) for image coding. *IEEE Trans. Comm., Vol. 46, No. 9*, pp 1215 – 1226.

S. Mohamed and M. Fahmy, 1995. Image compression using VQ-BTC. *IEEE Trans. Comm., Vol. 43, No. 7*, pp 2177 – 2182.

X. Muñoz, J. Freixenet, X. Cufí, J. Martí, 2003. Strategies for image segmentation combining region and boundary information. *Pattern Recognition Letters 24*, pp 375–392.

H. Musmann, P. Pirsch, and H. Grallert, 1985. Advances in picture coding. *Proc. IEEE Vol. 73, No. 4*, pp 523–548.

T. Nguyen and B. Oommen, 1997. Moment-preserving piecewise linear approximations of signals and images. *IEEE Trans. PAMI, Vol. 19, No. 1*, pp 84 – 91.

E. Ong, W. Lin, Z. Lu, S. Yao, and M. Etoh, 2004. Visual distortion assessment with emphasis on spatially transitional regions. *IEEE Trans. CAS-VT, Vol. 14, No. 4*, pp 559 – 566.

N. Otsu, 1979. A threshold selection method from gray level histograms. *IEEE Trans. Syst. Man Cybern.* SMC-9, pp 62–66.

D. Park, K. Nam, and R. Park, 1995. Multiresolution edge detection techniques. *Pattern Recognition, Vol. 28, No. 2*, pp 211–229.

W. Philips, 1991. Recursive computation of polynomial transform coefficients. *Elect. Lett., Vol. 27, No. 25*, pp 2337–2339.

J. Pinoli, 1997. The logarithmic image processing model: connections with human brightness perception and contrast estimators. *J. of Mathematical Imaging and Vision 7*, pp 341–358.

N. Ponomarenko, K. Egiazarian, V. Lukin, and J. Astola, 2007. High-quality DCT-based image compression using partition schemes. *IEEE Signal Processing Letters, Vol. 14, No. 2*, pp 105 - 108.

W. Pratt, 2001. Digital image processing. *John Wiley & Sons Inc., 3rd edition*, pp 30–32.

X. Ran and N. Farvardin, 1995. A perceptually motivated three-component image model-part II: Applications to image compression. *IEEE Trans. IP, Vol. 4, No. 4*, pp 430 – 447.

H. Ridder, 1996. Current issues and new techniques in visual quality assessment. *Proc. Int. Conf. on Image Processing, Vol. 1*, pp 869–872.

Y. Roterman and M. Porat, 2003. Progressive image coding using regional color correlation. *4th EURASIP Conference focused on Video/Image Processing and Multimedia Communications, Croatia*, pp 65 – 70.

P. Salembier and F. Marqués, 1999. Region-based representations of image and video: segmentation tools for multimedia services. *IEEE Trans circuits and systems for video technology, Vol. 9, No. 8*, pp 1147 – 1169.

P. Salembier and M. Pardas, 1994. Hierarchical morphological segmentation for image sequence coding. *IEEE Trans IP, Vol. 3, No. 5*, pp 639 – 651.

P Salembier, P. Brigger, J. Casas, and M. Pardas, 1996. Morphological operators for image and video compression. *IEEE Trans. IP, Vol. 5, No. 6*, pp 881 – 898.

D Sarkar, 1996. Boolean function-based approach for encoding of binary images. *Pattern Rec. Lett. 17*, pp 839–848.

M. Sendashonga and F. Labeau, 2006. Low complexity image quality assessment using frequency domain transforms. *IEEE Int. Conf. on Image Processing ICIP,* pp 385 – 388.

J. Shapiro, 1993. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. SP, Vol. 41, No. 12*, pp 3445 – 3462.

H. Sheikh, A. Bovik, and G. Veciana, 2005. An information fidelity criterion for image quality assessment using natural scene statistics. *IEEE Trans. IP, Vol. 14, No. 12*, pp 2117 – 2128.

H. Sheikh, M. Sabir, and A. Bovik, 2006. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Trans. IP, Vol. 15, No. 11*, pp 3441 – 3452.

Y. Shi and H. Sun, 2000. Image and video compression for multimedia engineering. *CRC Press*.

T. Sikora, 2005. Trends and perspectives in image and video coding. *Proc. IEEE, Vol. 93, No. 1*, pp 6 – 17.

S. Sinha and B. Schunck, 1992. A two stage algorithm for discontinuity preserving surface reconstruction. *IEEE Trans. PAMI, Vol. 14, No. 1*, pp 36 – 55.

J. Solé and P. Salembier, 2007. Generalized lifting prediction optimization applied to lossless image compression. *IEEE Signal Processing Letters, Vol. 14, No. 10*, pp 695 - 698.

H. Soliman and M. Omari, 2006. A neural networks approach to image data compression, *Applied Soft Computing 6*, pp 258–271.

P. Strobach, 1991. Quadtree-structured recursive plane decomposition coding of images. *IEEE Trans. SP, Vol. 39, No. 6*, pp 1380 – 1397.

R. Sunder, C. Eswaran, and N. Sriraam, 2006. Medical image compression using 3-D Hartley transform. *Computers in Biology and Medicine 36*, pp 958–973

T. Tran, 2000. The binDCT: fast multiplierless approximation of the DCT. *IEEE Signal Processing Letters, Vol. 7, No. 6*, pp 141 – 144.

G. Tsekouras, M. Antonios, C. Anagnostopoulos, D. Gavalas, and D. Economou, 2008. Improved batch fuzzy learning vector quantization for image compression. *Information Sciences 178*, pp 3895–3907.

C. Tseng, J. Hsieh, and J. Jeng, 2008. Fractal image compression using visual-based particle swarm optimization. *Image and Vision Computing 26*, pp 1154–1162.

Z. Wang and A. Bovic, 2002. A universal image quality index. *IEEE Sig. Proc. Lett., Vol. 9*, pp 81 – 84.

Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, 2004. Image quality assessment-from error visibility to structural similarity. *IEEE Trans. IP, Vol. 13, No. 4*, pp 600 – 612.

S. Wang, L. Kuo, H. Jong, and Z. Wu, 2005. Representing images using points on image surfaces. *IEEE Trans. IP, Vol. 14, No. 8*, pp, 1043 – 1056.

T. Watanabe, 1997. Picture coding employing B-spline surfaces with multiple vertices. *Elect. & Comm. in Japan Part I, Vol. 80, No. 2*, pp 55 – 65.

C. Wee, R. Paramesran, and R. Mukundan, 2007. Quality assessment of gaussian blurred images using symmetric geometric moments. *14th Int. Conf. on Image Analysis and Processing ICIAP*, pp 1 – 6.

D. Weken, M. Nachtegael, and E. Kerre, 2002. An overview of similarity measures for images. *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP 02, Vol.4,* pp IV-3317 – IV-3320.

D. Weken, M. Nachtegael, and E. Kerre, 2007. Combining neighbourhood-based and histogram similarity measures for the design of image quality measures. *Image and Vision Computing 25*, pp 184 – 195.

P. Wellimen, T. Reed, and M. Kunt, 1991. Image sequence coding by split and merge. *IEEE Trans. Comm, Vol. 39, No. 12*, pp 1845 – 1855.

R. Westwater and B. Furht, 1997. Real-time video compression. *Kluwer Academic*.

S. Winkler, 1999. Issues in vision modeling for perceptual video quality assessment. *Signal Processing 78*, pp 231 – 252.

B. Wohlberg and G. DeJager, 1999. A review of the fractal image coding literature. *IEEE Trans. IP, Vol. 8, No. 12*, pp 1716 – 1729.

X. Yang, H. Ren, and B. Li, 2008. Embedded zerotree wavelets coding based on adaptive fuzzy clustering for image compression. *Image and Vision Computing 26*, pp 812–819.

Y. Yemez, B. Sankur, and E. Anarim, 2005. An object-oriented video codec based on region growing motion segmentation. *Proc. Int. Conf. on Image Processing*, pp 444 – 447.

T. Yu, 1994. Nonlinear gradient-based edge detection algorithms in the telesign system. *IEEE Asia-Pacific Conference on Circuits and Systems*, pp 661–666.

Q. Yu and S. Xie, 2004. An image quality assessment method based on fuzzy inference rules. *8th Int. Conf. on Control, Automation, Robotics, and Vision, Kunming, China*, pp 702 – 705.

R. Zampolo and R. Seara, 2005. A comparison of image quality metric performances under practical conditions. *IEEE Int. Conf. on Image Processing ICIP, Vol. 3*, pp III 1192 – 1195.

G. Zhai, W. Zhang, X. Yang, and Y. Xu, 2005. Image quality assessment metrics based on multi-scale edge presentation. *Proc. IEEE Workshop on Signal Processing Systems Design and Implementation, SIPS*, pp 331 – 336.

G. Zhai, W. Zhang, X. Yang', S. Yao, and Y. Xu, 2006. GES: a new image quality assessment metric based on energy features in Gabor transform domain. *Proc. IEEE Int. Symposium on Circuits and Systems, ISCAS*, pp 1715 – 1718.

G. Zhai, W. Zhang, Y. Xu, and W. Lin, 2007. LGPS: Phase based image quality assessment metric. *IEEE Workshop on Signal Processing Systems,* pp 605 – 609.

Y. Zhang and D. Adjeroh, 2008. Prediction by partial approximate matching for lossless image compression. *IEEE Trans. IP, Vol. 17, No. 6*, pp 924 - 935.

W.Zhang, Y. Wang, and G. Hu, 2008. Compression of multi-polarimetric SAR intensity images based on 3D-matrix transform. *IET Image Process., Vol. 2, No. 4*, pp. 194–202.

Y. Zhou, D. Chen, C. Li, X. Li, and H. Feng, 2003. A practice of medical image quality evaluation. *IEEE Int. Conf. Neural Networks & Signal Processing, Nanjing, China*, pp 204 – 207.

H. Zhu and H. Wu, 2005. New paradigm for compressed image quality metric: exploring band similarity with CSF and mutual information. *Proc. IEEE Int. Geoscience and Remote Sensing Symposium, IGARSS 05, Vol. 2*, pp 696 – 699.

Y. Zhou, C. Zhang, and Z. Zhang, 2008. Fast hybrid fractal image compression using an image feature and neural network. *Chaos, Solitons and Fractals 37*, pp 623–631.

D. Ziou and S. Tabbone, 1998. Edge detection techniques – an overview. *Int. J. of Pattern Recognition and Image Analysis 8(4)*, pp 537–559.