3 IMAGE DATA COMPRESSION AND STORAGE

Prof. Hong Ren Wu[1], Dr. Damian M. Tan[1], Dr. Tom Weidong Cai[2] and Prof. David Dagan Feng [2,3]

[1] *School of Electrical and Computer Engineering, Science, Engineering & Technology Portfolio, RMIT University, Melbourne, Australia*
[2] *Biomedical & Multimedia Information Technology (BMIT) Research Group, School of Information Technologies, University of Sydney, Australia.*
[3] *Cnetre for Multimedia Signal Processing, Department of Electronic & Information Engineering, Hong Kong Polytechnic University*

CHAPTER 3 IMAGE DATA COMPRESSION AND STORAGE

3.1 INTRODUCTION

Picture compression is an important tool in the modern digital world. Over the years, the move towards the digital media has led to the proliferation of digital picture compression systems. This particularly noticeable in: the entertainment industry; consumer electronics; security/surveillance systems, etc. However, data compression is becoming increasingly important in biomedical imaging applications as well, due to increased popularity of digital biomedical imaging systems, the constant improvement of image resolution, and the practical need for on-line sharing of information though network. Picture compression came about following the advent of analogue television broadcasting. Initial methods to limiting signal bandwidth for transmission were relatively simple. They include sub-sampling to lower picture resolution and/or interlacing television pictures into alternate fields in alternate picture frames. With the introduction of colour television, sub-sampling was extended the colour channels as well [1]. The digital picture is seen as a natural migration from analogue picture. Hence, it is felt that digital picture compression is, in many ways, a natural evolution to analogue compression. While this is true in some sense, the nature of digital signals and that of analogue signals are quite different. Consequently, the methods for compressing digital and analogue pictures are distinct from one another. This chapter will first provide a basic introduction to digital picture compression focusing on general concepts and methods, then introduce some advanced data compression techniques. These techniques are used in noisy medical image data sets with high compression ratio and improved image quality, which have pioneered the biomedical diagnostically lossless data compression research. The extended discussion of the classical data compression can be found in [2-8], while the new research in diagnostically lossless data compression can be found in the references given in the later sections.

3.2 PICTURE COMPRESSION

A basic compression system has both an encoding and a decoding component. The encoder is responsible for compression while the decoder handles the reverse process, i.e., decompression. The goal of any compression system is to reduce the size of signal data while maintaining information integrity, or a certain degree of it. In the context of compression, it is important to note the difference between *data* and *information*. Data represents the individual sample of signal while information conveys the content of all data samples. From this perspective, compression can be seen as a function of data versus information as expressed by Shannon's theory of entropy [9]. Before moving on to information entropy, a brief overview is presented first of the concepts and terminologies in picture coding.

3.2.1 Picture Coding Concepts and Terminologies

Pictorial data in general is representative of both two-dimensional (image) and three-dimensional (video) spaces, but in biomedical domain, 3D volume images (e.g., CT and MRI images) and 4D time-varying 3D volume images (e.g., PET and SPECT images) are often used. In image processing terminology, an image is made up of individual *pixel* (picture element) arranged by rows and columns (or *voxel* for volume images arranged by rows and volumns for each image plane). Video data consists of multiple static images, commonly referred to as *frames*, arranged along the temporal/spatial[1] axis (Figure 3.1). Pixels in a picture have a common depth which determines the number of discrete luminance and/or chrominance levels. This pixel depth is measured in *bits* following the binary system utilized in existing computer architectures. Most common images have bit depths of 8-bits per pixel per channel while medical images may have between 10-bits to 16-bits per pixel. Each bit, within the continuous string of bits, is a binary number with value of 0 or 1. An 8-bit pixel therefore may range from $0000\ 0000_2$ to $1111\ 1111_2$ which in decimal terms is 0 to 255, having 2^8 (256) discrete levels. For any unsigned[2] binary number, the left most bit in a binary sequence is called the *most significant bit* (msb) having the highest order of magnitude. The right most bit is the *least significant bit* (lsb) with the lowest order of magnitude.

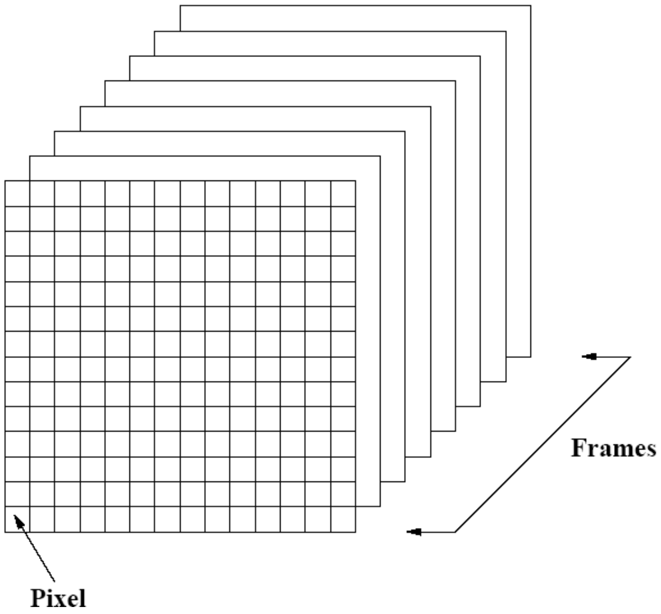


Figure 3.1: Illustration of pixels and frames

Natural pictures may be greyscale or coloured. Greyscale pictures have a single luminance channel while coloured pictures have three separate channels. Typically, colour channels are arranged according to the primary colours of red, blue and green (*RGB* colour space). However, it is also a common practice to represent colour channels in the component

[1] Some volumetric images such as biomedical images mentioned before are three dimensional spatial projection data or reconstructed projection data.

[2] Non-negative number.

colour domain with one luminance and two chrominance channels (YC_rC_b colour space) [1, 10].

The uncompressed representation of symbols/pixels[3] appears in the form of *fixed length binary codes*. The minimum length, l_m, of these codes is dependent on the size of a given symbol set[4], *n*, govern by

$$l_m = \lceil \log_2(n) \rceil . \tag{3.1}$$

Here, $\lceil \quad \rceil$ denotes rounding up to the nearest integer[5]. For digital pictures, this code length is dependent on pixel depth. For example, an 8-bit pixel uses an 8-bit fixed length code. Although compressed pictures are often quoted in compression ratios, the more conventional measurement is in *bitrate*, taken in units of *bits per pixel* (bpp). Compression ratio, C_{ratio}, defined as

$$R_c = \left(\frac{bpp_{uncompressed}}{bpp_{compressed}} \right) : 1 \tag{3.2}$$

is a measurement of compression gain where as bitrate is a measurement of the size of compressed data.

3.2.2 Shannon's Theory of Entropy

Shannon's theory of *entropy* [9], in short, stipulates that a minimum amount of data is necessary to represent a certain amount of information. If the amount of data falls short of the entropy, then information cannot be fully represented, leading to information loss. Conversely, if the entropy level is surpassed, then an excessive amount of data has been used to represent information, thus some redundancies exist. Given a symbol set $\boldsymbol{A}$, the discrete form of information entropy *H*, in bits, is written as

$$H = \sum_{i=1}^{n} p(i) \log_2 \left(\frac{1}{p(i)} \right) = -\sum_{i=1}^{n} p(i) \log_2 p(i) \ , \tag{3.3}$$

where *p(i)* if the probability associated with the occurrence of symbol $A(i)$ such that $\sum p(i) = 1$, for *i* = {*1,2,…,n*} and $A(i) \in \boldsymbol{A}$. It is clear from (3.3) that *H* is a summation of all the information contained within each symbol in $\boldsymbol{A}$, relative to the probability of occurrence. This self-contained information in each symbol is referred to as *self-information* and is defined as

[3] Symbol is a reference to generic data (e.g., text, audio, etc.) where as pixel is specifically associated with picture data.

[4] The number of different symbols. For pixels, it is the number of discrete magnitude levels.

[5] Rounding is needed to remove fractions since the fixed length binary codes have integer representation (quantization).

$$\iota(i) = -\log_2 p(i) \tag{3.4}$$

Redundancy in data, as defined by Shannon [9], is a proportional difference between the entropy, H, and the actual size of the source data, l_m as defined in (3.1), relative to 1. It is formulated as

$$\rho = 1 - \left(\frac{H}{l_m}\right) \tag{3.5}$$

i	**1**	**2**	**3**	**4**	**5**	**6**
***A*(*i*)**	***a***	***b***	***c***	***d***	***e***	***f***
***P*(*i*)**	0.3	0.1	0.2	0.05	0.15	0.2

Table 3.1: Example probability distribution of symbol set ***A***.

Example 1: *Given a set of symbols,* ***A****, with the probability distribution shown in Table 3.1, the entropy is computed as:*

$$\begin{aligned} H(\boldsymbol{A}) &= -\left[p(1)\cdot\log_2(p(1)) + p(2)\cdot\log_2(p(2)) + \ldots + p(6)\cdot\log_2(p(6))\right] \\ H(\boldsymbol{A}) &= -\left[0.3\cdot\log_2(0.3) + 0.1\cdot\log_2(0.1) + \ldots + 0.2\cdot\log_2(0.2)\right] \\ H(\boldsymbol{A}) &= 2.4087 \quad \text{bits per symbol} \end{aligned}$$

For most uncompressed natural audio and picture signals, the amount of data used to represent the information is above the entropy threshold. Consequently, it is possible to compress these signals. For example, according to (3.1), the symbol set ***A*** from Table 3.1, with symbol size of six (n=6), required 3 bits to denote each symbol with fixed length codes. However, the information entropy associated with ***A*** is 2.4087 bits per symbol. From (3.5), the redundancy in ***A*** is calculated as

$$\begin{aligned} \rho(\boldsymbol{A}) &= 1 - \left(\frac{2.4087}{3}\right) \\ &= 0.1971 \end{aligned}$$

3.2.3 Entropy Coding

Entropy coding is a reference term for information lossless compression at or near the data entropy. Its general operation consists of two stages: modelling and coding. Modelling is performed to identify and describe data redundancies. It is carried out through statistical analysis of data set in order to capture the *probability distribution of symbols* (PDS) in the data set. The coding phase then encodes the information in data, based on the description of data derived during the modelling phase, by assigning a distinct code to each respective symbol[6]. The size of each code, measured in number of bits, is dependent on the probability of occurrence of its respective symbol. Generally, the most common symbol will have the smallest code size while the least common symbol would have the largest code size. The manner in which these codes are generated is dependent on the coding algorithm. There is a variety of entropy coding algorithms available [8]. The most common of these is the Huffman code [11], which generates its alphabet of codeword through recursive sorting of source symbols, of a given probability distribution, into a binary tree. In each iteration, the Huffman coding algorithm performs the following operations:

i. Sort the PDS in descending order.

ii. The two symbols with the lowest probability of occurrence are then grouped together to build a tree branch and generate an updated symbol table. For every tree branch, each of the two merged symbol is assigned a binary digit, 0 for the last and 1 for the second last. With each successive iteration, the merged symbols will then progressively build up their binary codeword (BCW).

iii. Iterate process until the root of the tree is reached. This occurs when no more tree branches can be formed, i.e., when there is only one symbol remaining. The number of iterations, *i*, required to generate the entire tree is one less than the total number of symbols, *n*. i.e., $i = n-1$.

Example 2: *Given the PDS in Table 3.1, the Huffman code is generated as follows:*

Iteration 1:

probability: $p(a) = 0.3$, $p(b) = 0.1$, $p(c) = 0.2$, $p(d) = 0.05$, $p(e) = 0.15$, $p(f) = 0.2$

i. Sorting symbols:

$$A = \{a, b, c, d, e, f\} \Rightarrow \{a, c, f, e, b, d\}$$

ii. Group lowest two symbols and build tree branch

$z = \{b,d\}$, $b=1$, $d=0$,

$A_1 = \{a, c, f, e, z\}$

binary codeword: $b=1$, $d=0$

iii. $\text{size}(A_1) > 1$

Iteration 2:

probability: $p(a) = 0.3$, $p(c) = 0.2$, $p(f) = 0.2$, $p(e) = 0.15$, $p(z) = 0.15$

i. Sorting symbols:

$$A_1 = \{a, b, c, d, z\} \Rightarrow \{a, c, f, e, z\}$$

ii. Group lowest two symbols and build tree branch

$y = \{e,z\}$, $e=1$, $z=0$,

$A_2 = \{a, c, f, y\}$

binary codeword: $e=1$, $b=01$, $d=00$

[6] Codes may also be assigned to a sequence of symbols as in the case of arithmetic coding.

iii. size(A_2) > 1

Iteration 3:

probability: $p(a) = 0.3$, $p(c) = 0.2$, $p(f) = 0.2$, $p(y) = 0.3$

i. Sorting symbols:

$$A_2 = \{a, c, f, y\} \Rightarrow \{a, y, c, f\}$$

ii. Group lowest two symbols and build tree branch

$x = \{c,f\}$, *c=1, f=0,*

$$A_3 = \{a, y, x\}$$

binary codeword: *c=1, f=0, e=1, b=01, d=00*

iii. size(A_3) > 1

Iteration 4:

probability: $p(a) = 0.3$, $p(y) = 0.3$, $p(x) = 0.4$

i. Sorting symbols:

$$A_3 = \{a, y, x\} \Rightarrow \{x, a, y\}$$

ii. Group lowest two symbols and build tree branch

$w = \{a,y\}$, *a=1, y=0,*

$$A_4 = \{x, w\}$$

binary codeword: a*=1, c=1, f=0, e=01, b=001, d=000*

iii. size(A_4) > 1

Iteration 5:

probability: $p(x) = 0.4$, $p(w) = 0.6$

i. Sorting symbols:

$$A_4 = \{x, w\} \Rightarrow \{w, x\}$$

ii. Group lowest two symbols and build tree branch

$v = \{w,x\}$, *w=1, x=0,*

$$A_5 = \{v\}$$

binary codeword: *a=11, c=01, f=00, e=101, b=1001, d=1000*

iii. size(A_5) = 1: Stop Iteration

This operation is visualized in Figure 3.2.

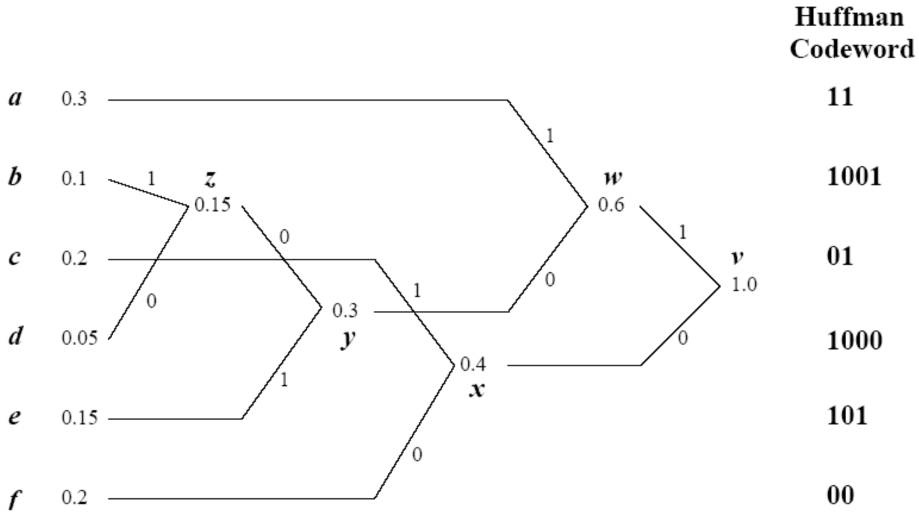


Figure 3.2: Huffman tree corresponding to the PDS in Table 3.1. The Huffman codeword are generated by tracing the binary numbers along each branch from the root to each leaf of the tree.

It is important to note that each individual code generated in Figure 3.2 is distinct and therefore uniquely decodable. For example, the encoded binary sequence **000111011000** can only be decoded into symbols *f*(**00**), *c*(**01**), *a*(**11**), *c*(**01**), *d*(**1000**).

The compression rate, r_H, associated with the Huffman codes in Figure 3.2 can be computed as follows:

$$r_H = \sum_{i}^{N} p(i) \cdot b(i) \tag{3.6}$$

where $p(i)$ and $b(i)$ are respectively the probability of occurrence and the codeword length of the i^{th} Huffman codeword. The expected compression rate of the Huffman codes in Figure 3.2 is

$$r_H = \left[p(1)\cdot b(1) + p(2)\cdot b(2) + ... + p(6)\cdot b(6)\right]$$
$$r_H = [0.3\times 2 + 0.1\times 4 + 0.2\times 2 + 0.05\times 4 + 0.15\times 2 + 0.2\times 3]$$
$$r_H = 2.45 \text{ bits per symbol}$$

The Huffman code is generally quite efficient, compression wise. It also has simple and fast implementation. However, its integer codeword may be less efficient in situations where fractional codeword[7] occurs. Take the example of symbol set ***A*** (Table 3.1). A

[7] Codeword with fractional part.

comparison of its Huffman code size, S_{hc}, and its optimum code size, $S_{oc} = \log_2(\boldsymbol{A})$, given in Table 3.2, demonstrates some inefficiencies of the Huffman code for symbols ***a***, ***c*** and ***f***. While the Huffman codes for symbols ***b***, ***d*** and ***e*** are more efficient than the optimal code, these symbols appear with less frequency than ***a***, ***c*** and ***f***. Therefore, on average, the Huffman codes will require 2.45 bits to code one symbol where as the optimum codes[8] require 2.4087 bits per symbol.

i	**1**	**2**	**3**	**4**	**5**	**6**
***A*(*i*)**	***a***	***b***	***c***	***d***	***e***	***f***
$S_{oc}(i)$	1.737	3.3219	2.3219	4.3219	2.737	2.3219
$S_{hc}(i)$	2	4	2	4	3	2

Table 3.2: Symbol set ***A***. Codeword sizes (bps) for the optimum code (S_{oc}) and the Huffman code (S_{hc}). The Huffman code size is shown in Figure 3.2.

In contrast to Huffman coding, arithmetic coding [12, 13] can indirectly handle fractional codeword and consequently has become the preferred choice for entropy coding in a number of applications [14, 15]. Instead of encoding each symbol separately, arithmetic coding encodes a group of symbols. The arithmetic codeword used to represent a sequence of symbols is formed through successive cascades of the probability interval of source symbols. Figure 3.3 illustrates the formation of the cascaded intervals. A step by step approach for encoding and decoding is shown in Tables 3.3 and 3.4 respectively.

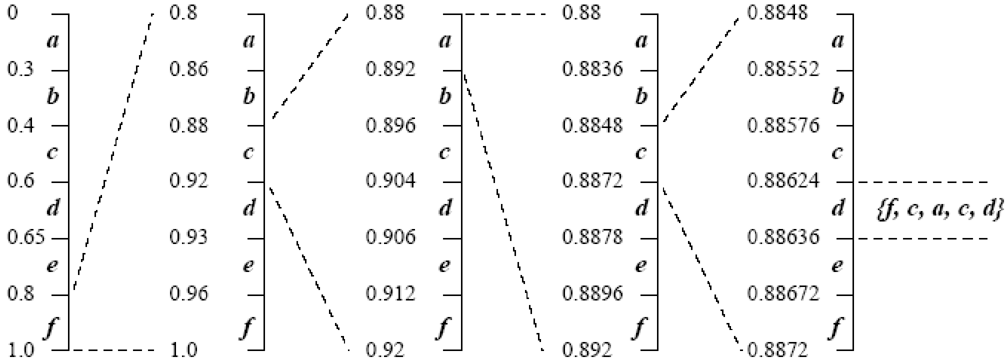


Figure 3.3: Arithmetic coding with cascaded intervals for symbols {*f*, *c*, *a*, *c*, *d*} from PDS in Table 3.1.

The encoding procedure for arithmetic coding has two steps in each iteration:

i. Identity the interval range, *R*, of symbol to be coded,

$$R = H - L \tag{3.7}$$

where *H* and *L* are the upper and lower bound of the interval range, respectively.

[8] Optimum code encodes at the entropy level.

ii. Update both the upper and the lower bounds relative to the encoded symbol

$$H = L + R \times S_H \tag{3.8}$$

$$L = L + R \times S_L \tag{3.9}$$

where S_H and S_L are the upper and lower interval bounds of the encoded symbol.

symbol	*R*	*L*	*H*
	1.0	0.0	1.0
f	0.2	0.8	1.0
c	0.04	0.88	0.92
a	0.012	0.88	0.892
c	0.0024	0.8848	0.8872
d	0.00012	0.88624	0.88636

Table 3.3: Arithmetic encoding of symbols sequence {*f*, *c*, *a*, *c*, *d*} from PDS in Table 3.1.

The decoding phase is also carried out in two steps per symbol:

i. Locate the symbol interval for codeword, C_a: $S_L < C_a < S_H$

ii. Update the codeword for decoding the next symbol

$$C_a = (C_a - S_L) / R_s \tag{3.10}$$

where $R_s = S_H - S_L$ is the symbol range.

code	symbol	S_L	S_H	R_s
0.88625	***f***	0.8	1.0	0.2
0.43125	**c**	0.4	0.6	0.2
0.15625	***a***	0.0	0.3	0.3
0.52083	**c**	0.4	0.6	0.2
0.60417	***d***	0.6	0.65	0.05

Table 3.4: Arithmetic decoding of symbols sequence {*f*, *c*, *a*, *c*, *d*} from PDS in Table 3.1.

The initial coding intervals are taken as the boundary intervals of the PDS. As each symbol is encoded, the coding intervals are refined. These refined intervals are proportional to the PDS, but limited to within the interval range of the coded symbol. For example, in Figure 3.3 (also Table 3.3), after the first symbol (***f***) is encoded, the intervals for the proportional distribution of symbols are limited within the range of 0.8 and 1.0. Encoding the symbol sequence {***f***, ***c***, ***a***, ***c***, ***d***} in Figure 3.3 (Table 3.3) will result in the final coding interval (0.88624, 0.88636). Any arithmetic code number, C_a, that resides within this range can be used to represent the encoded symbol sequence. Generating the cascaded probability intervals is a trivial task. The real challenge for arithmetic coding is how to efficiently encode the number between two given interval points. With each successive cascade, the range of the probability interval decreases. As a result, the arithmetic precision needed to represent the code number increases. This in turn dictates the number of bits needed to encode the arithmetic code number. In arithmetic coding, there is no fixed codeword for any particular source symbol, since symbols are coded in groups rather than individually. Furthermore, as a consequence of group encoding, individual symbol may be coded indirectly in fractions. Take the example in Figure 3.3, the symbol sequence {***f, c, a, c, d***} may be represented by the code number C_a = 0.88628. Its 8-bit integer representation is 226 ($(2^8 - 1) \times 0.88628$)[9]. Since an 8-bit codeword[10] is used to encode five symbols, it is only possible to extrapolate the average code size per symbol, l_a, by

$$l_a = \frac{\text{codeword length}}{\text{no. of symbols}} = \frac{8}{5} = 1.6 \text{ bits per symbol}$$

However, this does not imply that only 1.6 bpp is needed to encode symbols with the probability distribution given in Table 3.1. The example given above, taken from Figure 3.3, only covers the first five symbol. In addition, the integer representation of the arithmetic code number, C_a, is an overly simplistic view of arithmetic coding. The full description of arithmetic coding is given in [110]. A general comparison between arithmetic and Huffman coding is covered in [8].

Entropy coding may operate in an *adaptive* or *static* manner [8]. The difference between these two modes is that the former is able to adapt to changes in data while the latter does not. Adaptive entropy coding, also referred to as dynamic entropy coding, utilizes a pre-determine PDS in its initial operation. This initial PDS is generally acquired, heuristically, through generic sample data. During the encoding process, the PDS is continually updated with coded data to reflect the statistical profile of the data set more accurately. Adaptive coding is particularly useful in instances where the a priori profile of the PDS is unknown[11] or if different segments of the source data exhibit different PDS profile. In addition to this, adaptive coding is often employed in situations where the statistical analysis of data is impractical due time constrains. This is envisaged when dealing with volumetric data. In instances where data volume or time constraints are not an issue, then static entropy coding may be employed. Static entropy coding generally models each data set independently to capture the exact PDS. However, coding with the exact PDS incurs some overheads since the PDS used for encoding is also needed for decoding. For example, in order to decipher the Huffman codes at the decoding end, a copy of the Huffman tree generated at the encoding end must be made available. This is only possible if the PDS is

[9] An 8-bit binary integer number has 256 discrete level ranging from 0 to 255 ($2^8 - 1 = 255$).

[10] In practice, the codeword size must be selected in a fashion which allows distinct representation of all possible combinations of symbol sequences. Here, an 8-bit codeword is only used as an example to illustrate fractional codes.

[11] For example, real-time encoding where data is encoded as soon as it is digitized.

sent to the decoder. Consequently, an overhead is incurred for storing/transmitting the PDS needed to generate the Huffman tree.

While the entropy for any given set of data may be determined analytically, it is understood that, under normal circumstances, existing entropy coding techniques can never reach the entropy threshold. For static entropy coding, this is due to the practical limitation of compression overheads. In the case of adaptive entropy coding, transmission of the PDS is unnecessary since a default PDS already exist. However, for adaptive coding, the problem lies in the adaptation process which infers inefficiency, hence the need to adjust. Though the adaptation approaches the entropy, it would never reach the entropy threshold since the entropy is progressively estimated from causal data samples, not all data samples. Coding at the entropy threshold is only possible if there is no variation in data statistics and if the initial PDS is optimal, a most unusual situation if it does occur.

3.2.4 Classification of Picture Compression

Picture coding has traditionally been classified into two general categories: *Lossless* and *Lossy*. Lossless coding maintains information integrity. Lossy coding in contrast degrades information integrity to a certain extent. However, any information lost is offset by higher compression ratio. Therefore, lossy compression is a balancing act between information quality and compression performance as dictated by the *Rate versus Distortion* (R-D) curves of pictures (Figure 3.4). Ideally, lossless compression would be the approach of choice for encoding picture data. In practice, however, it is unable to deliver the necessary compression ratio required for most consumer applications. The limitation of lossless compression becomes obvious when dealing with video or volumetric data. Consequently, there is preference towards lossy coding of digital video as reflected in existing industrial standards [16].

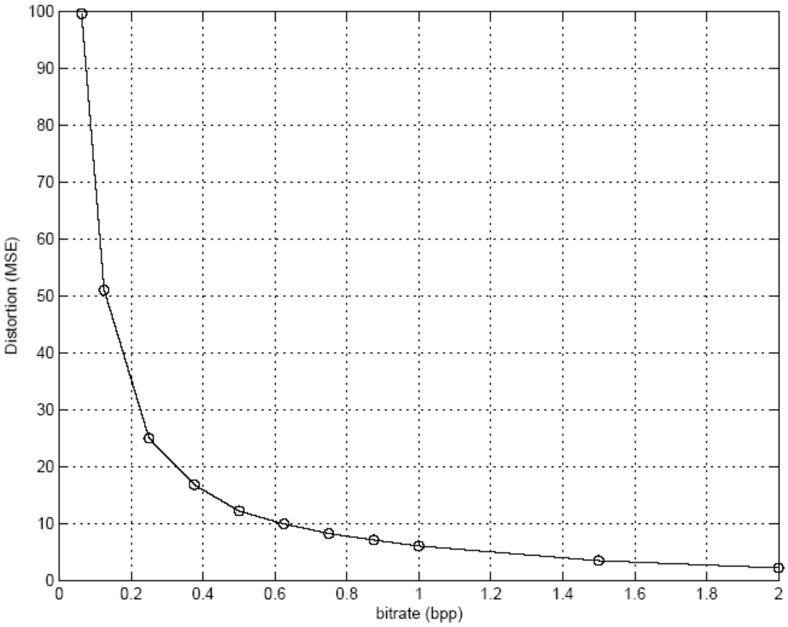


Figure 3.4: Example R-D curve (*top*) for the *Lena* image (*bottom*) coded at different bitrates with the JPEG2000 coder [15]. As the bitrate increases, the distortion decreases. An increase in bitrate corresponds to lower compression ratio.

3.2.5 Lossless Picture Coding

Lossless picture coding in it most basic form is equivalent to entropy coding. However, picture data being two-dimensional in nature, generally has strong correlation between adjacent pixels. Consequently, it is common practice to employ predictive coding prior to entropy coding to further improve compression performance. Utilizing predictive coding such as *Differential Pulse Code Modulation* (DPCM)[12] [17, 18] has the effect of reshaping of the PDS. For natural images, predictive coding leads to sharper PDS usually centred about the zero prediction with a Gaussian or a Laplacian profile. This translates to better compaction of data since the probability distribution is concentrated on fewer numbers of symbols as evident in Figure 3.5.

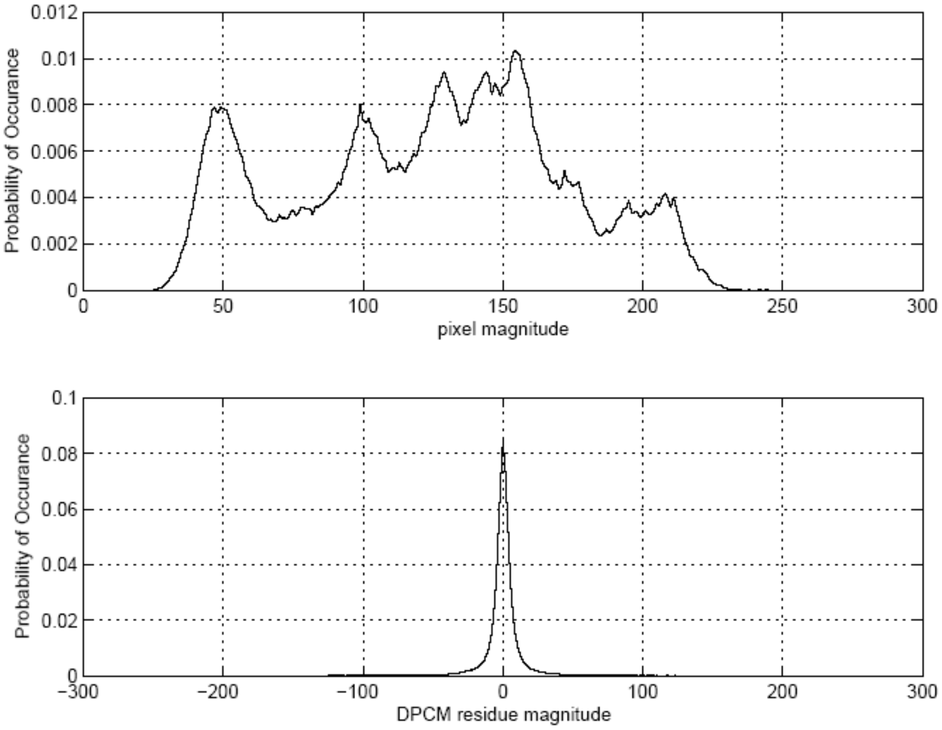


Figure 3.5: PDS of the *Lena* image. *Top*: PCM (entropy - 7.4456 bpp). *Bottom*: row wise DPCM (entropy - 5.0475 bpp).

A simple DPCM encodes the difference between two pixels such that

$$r[n] = x[n] - x[n-1] \tag{3.11}$$

[12] *Pulse Code Modulation* (PCM) [19] encodes analogue signals in digital waveforms.

where *r*[n] is the predicted *residue*, *x*[n] and *x*[n-1] are the current and previous pixels, respectively. It is however more common in predictive coding to use multiple *causal* samples to generate the predicted residue. Causality in this context refers to samples which have been encoded, i.e., *x*[i] for i < n. The convention in image coding is to read pixels left to right and top to bottom. Thus any pixel that is above or to the left of the current pixel is considered causal (Figure 3.6). One common approach to predictive coding is to utilize all four immediate neighbouring causal samples [20]. Other combinations that extend beyond the immediate neighbours are also possible [14].

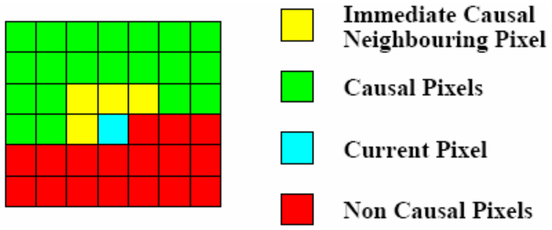


Figure 3.6: Causal samples for predictive coding

Generally, having more causal pixels would improve prediction since correlation between adjacent pixels or pixels in close proximity is typically high. However, not all adjacent pixels are correlated. Further, pixel correlation degrades relative to distance as evident in Figure 3.7.

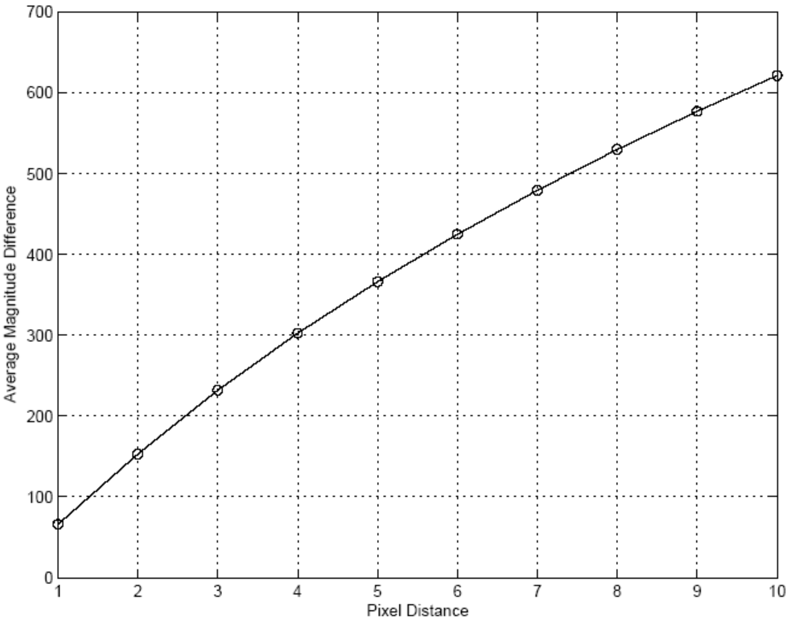


Figure 3.7: Correlation relative to pixel distance measured according to the average difference between pixels of the *Lena* image.

3.2.5.1 Context Based Coding

In text compression, it is simple to envisage that the occurrence of a letter in an English word[13] is dependent on previous letters in that word [21]. For example, if the letter 'Q' was encountered, the probability of the next letter being 'U' is almost certain. Similarly, if the two previous letters in a word is 'QU', it is likely that the next letter would be a vowel. In light of this, it would be most prudent to adjust the coding operation to take advantage of this conditional correlation. Context based coding operates by selecting the most suitable context for encoding data based on the behaviour of past data samples[14]. Each context within a context coder maintains a separate PDS, independent of other contexts, for entropy coding [8]. For image compression, context selection is dependent on structural characteristics of neighbouring pixels. Typically, these characteristics are measured in terms of horizontal, vertical and/or diagonal edges. Context coding employs multiple conditional predictors for residue coding. Again, the predictor selection for coding any given pixel is also dependent on structural characteristics. However, it is important to distinguish the difference between context selection and predictor selection. The purpose of context selection is to match the actual residue produced by the selected predictor to a suitable probability distribution. The predictor on the other hand, is selected in order to minimize the prediction residue. Having multiple predictor increases computation complexity and cost. However, it usually leads to better prediction, hence a more favourable probability distribution for entropy coding [8]. The manner in which context coding is employed varies with different lossless algorithms. This is illustrated in the next two sections with two different coders: LOCO and CALIC.

3.2.5.2 Low Complexity (LOCO) Coder

The LOCO coder [20] is the core of the JPEG-LS [22] coding standard. It relies on the four causal neighbouring pixels for prediction and employs a conditional predictor given by

$$\hat{x} = \begin{cases} \min(W,N), & NW \geq \max(W,N) \\ \max(W,N), & NW \leq \min(W,N) \\ W + N - NW, & \textit{otherwise} \end{cases} \tag{3.12}$$

where $\hat{x}$ is the predicted pixel with *W*, *N* and *NW* the neighbouring pixels illustrated in Figure 3.8. The residue is the difference between the original pixel, *x*, and the predicted pixel, $\hat{x}$.

[13] In so far as the natural English language is concerned.

[14] It is perhaps most accurate to differentiate between multi-context and uni-context coding, as opposed to context and non-context coding, since all coding schemes has at least one context by default.

		NN	*NNE*
	NW	*N*	*NE*
WW	*W*	*x*	

Figure 3.8: Neighbouring samples used for prediction for LOCO and CALIC coders.

The context selection process is dependent on three difference measures, D_i for $i = \{1, 2, 3\}$, with simple horizontal and vertical edge detection. These difference measures are defined as

$$\begin{aligned} D_1 &= NE - N \\ D_2 &= N - NW \\ D_3 &= NW - W \end{aligned} \tag{3.13}$$

The response of each difference measure is partitioned into 2*T*+1 equal distance intervals, z_{ij}, such that

$$z_i = \begin{cases} -T, & D_i \leq -T \\ -T+1, & -T < D_i \leq -T+1 \\ \ldots & \\ 0, & D_i = 0 \\ \ldots & \\ T-1, & T-1 < D_i \leq T \\ T, & T < D_i \end{cases} \tag{3.14}$$

The equal distance interval is chosen to simplify implementation. Ideally however, the intervals should be optimized according to structural statistics of images. The total number of context, *C*, is then taken as the number of possible permutations for the given numbers of intervals and difference measures, $C = (2T+1)^3$. The number of context may be reduced by merging symmetric intervals in (3.14) such that $-T \leq z_i \leq T \rightarrow 0 \leq |z_i| \leq T$ and thus resulting $C_{sym} = ((2T+1)^3 + 1)/2$ contexts. The LOCO coder encodes the residue with Golomb codes [23] under regular operation. However, if flat regions are encountered, adaptive run-length coding[15] is used.

[15] See [24] for a description of run-length coding.

3.2.5.3 CALIC

The Context Arithmetic Lossless Image Coder (CALIC) [14] is moderately sophisticated. Compared with the LOCO coder, it has a more developed edge detection, residue prediction and context modelling functions. Coding operation begins with two edge-sensitive gradient estimation functions based on neighbouring pixels (Figure 3.8)

$$\begin{aligned} d_h &= |W - WW| + |N - NW| + |NE - N| \\ d_v &= |W - NW| + |N - NN| + |NE - NNE| \end{aligned} \tag{3.15}$$

These gradient estimates determine the predicted pixel, $\bar{x}$, under the following conditions

$$\bar{x} = \begin{cases} N, & d_h - d_v > 80 \\ W, & d_v - d_h > 80 \\ (\tilde{x} + N)/2, & d_h - d_v > 32 \\ (\tilde{x} + W)/2, & d_v - d_h > 32 \\ (3\tilde{x} + N)/4, & d_h - d_v > 8 \\ (3\tilde{x} + W)/4, & d_v - d_h > 8 \end{cases}, \tag{3.16}$$

where

$$\tilde{x} = (N + W)/2 + (NE - NW)/4\,. \tag{3.17}$$

Context modelling is based on prediction error and texture. The prediction error, *e*, is modelled as

$$e = ad_h + bd_v + c|e_w| \tag{3.18}$$

where $e_w = W - \overline{W}$ is the previous prediction error with $\overline{W}$ being the prediction of W. The coefficients *a*, *b* and *c* are parameters to be optimized. For efficient operation, *a*=*b*=1 and $c = 2$ has been suggested [14]. In the basic CALIC implementation, *e* is optimally quantized into eight intervals. The boundaries of these intervals were obtained through dynamic programming and found to be $\xi_i = \{5,15,25,42,60,85,140\}$, $0 \le i \le 7$. The actual interval is $\phi = \{0 < e \le \xi_1, \xi_1 < e \le \xi_2, ..., \xi_6 < e \le \xi_7, \xi_7 < e\}$.

Texture context, being dependent on the activity of neighbouring pixels, is modelled under eight separate factors, C_t, defined as

$$C_t = \{x_0, x_1, ..., x_6, x_7\} = \{N, W, NW, NE, NN, WW, 2N - NN, 2W - WW\} \tag{3.19}$$

where $t = \{1,2,...,7\}$. Individual factor, x_k, is measured against the prediction derived in (3.12) as follows

$$b_k = \begin{cases} 0 & , x_k \geq \bar{x} \\ 1 & , x_k < \bar{x} \end{cases}, \qquad k = \{0,1,\ldots,7\}. \tag{3.20}$$

The combined measurement of all factors, $B = \{b_7, b_6, \ldots, b_0\}$, determines the behaviour of the pixel neighbourhood and forms number of texture contexts, $2^8 = 256$. However, due to dependencies[16] between factors in C_t, the actual number of texture context is reduced to 144. The overall number of context is a combination between texture and prediction error. For the basic CALIC implementation, the prediction error context is reduced from 8 to 4 to form a total of 144×4 = 576 coding contexts. Once the context has been determined through (3.18) and (3.19), the residue is entropy coded with arithmetic coding.

3.2.5.4 Near-Lossless Compression

A major limitation of lossless compression has always been its compression performance. To address this deficiency, the near-lossless coding scheme has been proposed [20]. Near-lossless coding allows for control degradation of picture quality in order to improve compression gain. This is carried out through the quantization of the prediction residue, *r*, prior to entropy coding[17]. This quantization operation is performed in a manner which guarantees that the quantized residue will not deviate beyond a certain point, $\pm\Delta$, as defined by users. The quantized residue, $\bar{r}$, is derived as follows

$$\bar{r} = \mathrm{sign}(r)\left\lfloor \frac{|r| + \Delta}{2\Delta + 1} \right\rfloor, \tag{3.21}$$

where Δ determines the maximum magnitude of deviation and sign(*r*) is defined as

$$\mathrm{sign}(r) = \begin{cases} -1 & , r < 0 \\ 1 & , r \geq 0 \end{cases}. \tag{3.22}$$

Since near-lossless coding scheme only appends an additional quantization stage to the coding system, it could be readily adapted to practically any lossless coding algorithms.

3.2.6 Transform Based Lossy Picture Coding

[16] See [14] for further details.

[17] Only the quantized residue, $\bar{r}$, is entropy coded, not the residue, *r*.

In order to attain higher compression ratio that surpasses that of lossless coding schemes, it has become a necessity to tolerate some information loss. Transform based coding is employed generally for this purpose. It consists of a transformation operator followed by quantization and bitstream coding (Figure 3.9).

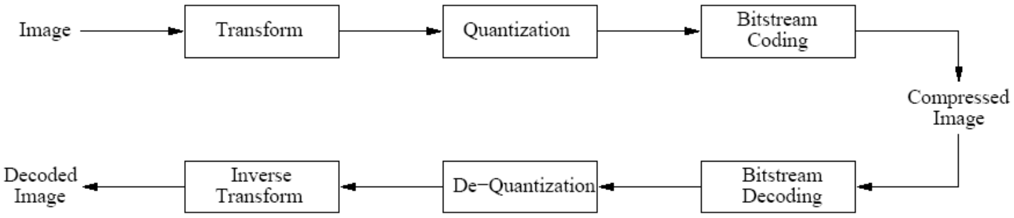


Figure 3.9: Principle components of transform based coding structure.

3.2.6.1 Transformation

The transformation operation is intended to rearrange data, in pictures, in a manner that facilitates compression. The most common form of data transformation applied in picture coding is frequency based. Other non-frequency transforms such as fractal [25, 26] have also been studied. A frequency transform project data from the time domain, **x**, to the frequency domain, **X**, relative to a given filter set, *s*,

$$\mathbf{X} = T_s(\mathbf{x}) \tag{3.23}$$

The transformation process, *T*, is carried out through a convolution operator ($*$) defined by

$$\mathbf{X} = T_s(\mathbf{x}) = \mathbf{S} * \mathbf{x} = \sum_{i}^{I} \mathrm{S}[i] \cdot \mathbf{x}[n-1] \tag{3.24}$$

where *S* is the transform filter of length *I*, *n* = {1,2,…,N} and *N* the number of data samples. In regards to filter length, most signal processing applications, including picture compression, typically employ *Finite Impulse Response* (FIR) filters. FIR filters have a finite number of filter taps, i.e., a finite filter length. This is desirable for practicability. The alternative *Infinite Impulse Response* (IIR) filters, while superior to FIR in terms of frequency selectivity [27], require infinite sampling. Infinite sampling may be carried out through recursive filtering [28, 29]. However, for invertible systems with forward (*analysis*) and inverse (*synthesis*) transformation, the synthesis IIR operation can only be realized with signals of finite samples[18] [27].

[18] e.g., still images.

The process of frequency transformation decomposes pictorial data into different subband images. In this respect, frequency transform is referred to as subband transform. Subband transforms are classified according to their decomposition structure and filters. The decomposition structure determines the way in which transform coefficients are arranged. This arrangement may be subband [30], block-based [31] or hierarchical [32, 33] in nature (Figure 3.10). The hierarchical decomposition is also known as multi-resolution or wavelet transform.

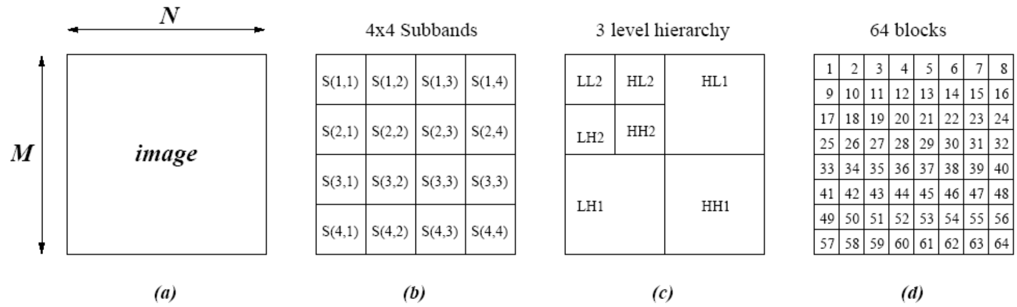


Figure 3.10: Transform decomposition structures. (a) An image of $M \times N$ dimension. (b) A 4×4 subband decomposition. Each band $S(k,l)$ with k, $l = \{1,2,3,4\}$ has $M/4 \times N/4$ transform coefficients. (c) A 2 level dyadic hierarchical decomposition. Each successive level is a quarter the resolution of the previous. (d) Block based transform with 64 blocks. Each block is of $M/8$ x $N/8$ dimension.

Image transformation is generally two-dimensional (2-D) since images are 2-D data. For video, there is a natural extension from 2-D to 3-D transform to account for the temporal dimension. However, due to the prevailing coding philosophy [10, 16], most video coders adhere to the 2-D transform for individual picture frame.

Filters are the core of the transform and dictate transformation characteristics such as frequency response. The principles of filter design originated from Fourier analysis [34]. Consequently, a basic and commonly employed tool in signal analysis is the Discrete Fourier Transform (DFT) [31] defined as

$$\begin{aligned} X_{DFT}[k] &= \sum_{m=0}^{M-1} x[m] \cdot e^{-\frac{2\pi j}{M}km} \\ &= \sum_{m=0}^{M-1} x[m] \cdot \left(\cos\left[\frac{\pi}{M}\left(m + \frac{1}{2} \right)k \right] - j \sin\left[\frac{\pi}{M}\left(m + \frac{1}{2} \right)k \right] \right) \end{aligned} \tag{3.25}$$

where M is the filter length, j the imaginary unit and k the frequency band of the selected filter. In picture compression, the application and performance of various transformation kernels have been explored [2, 4]. The more popular amongst these is the *Discrete Cosine Transform* (DCT) [31] formulated as

$$X_{DCT}[k]=\sum_{m=0}^{M-1}x[m]\cdot\cos\left[\frac{\pi}{M}\left(m+\frac{1}{2}\right)k\right] \tag{3.26}$$

The DCT is derivation of the real component of the DFT as evident in (20). It offers near optimal decorrelation performance second to that of the Karhunen-Loève Transform (KLT) [35], with respect to the first order Markov random process [7]. Decorrelation has been seen as an important feature for filters, in the past, since it corresponds to energy packing ability. Energy packing leads to compaction of statistical redundancy whereby a substantial amount of pixel energy is contained with a small number of transform coefficients. The popularity of the DCT is, in no small part, credited to the availability of fast DCT algorithms that has proliferated its practical application in digital communication equipment and services.

Separable filters such as the DFT and the DCT may be extended from one-dimension (1-D), as in (3.25) and (3.26), to two-dimension (2-D) through separate horizontal (φ_h) and vertical (φ_v) transformation operation performed in two stages [2, 27]. For the DCT, this is formalized as

$$X_{DCT}[k,l]=\sum_{m=0}^{M-1}\varphi_v(k,m)\cdot\sum_{n=0}^{N-1}x[k,l]\cdot\varphi_h(l,n) \tag{3.27}$$

with

$$\varphi_v(k,m)=\cos\left[\frac{\pi}{M}\left(m+\frac{1}{2}\right)k\right] \tag{3.28}$$

$$\varphi_h(l,n)=\cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)l\right]. \tag{3.29}$$

Filters are designed to address specific characteristic of the data they operate upon. They have various properties which are discussed in detail in [27]. For picture coding purposes, there are a number of highly desirable properties that filters should have such as phase linearity, orthogonality, etc. *Perfect reconstruction*, defined as

$$\mathbf{X}=T^{-1}(T(\mathbf{x})) \tag{3.30}$$

where x is the input signal with T and T^{-1} the forward and reverse transform, respectively, allows for complete recovery of data. Invertibility, while not strictly necessary for lossy coding, has become desirable in light of the move towards *scalable* coding [36]. For full scalability in picture quality, picture coders must have the flexibility of encoding from lossy to lossless quality such as in the case of a JPEG2000 compliant image coder [15].

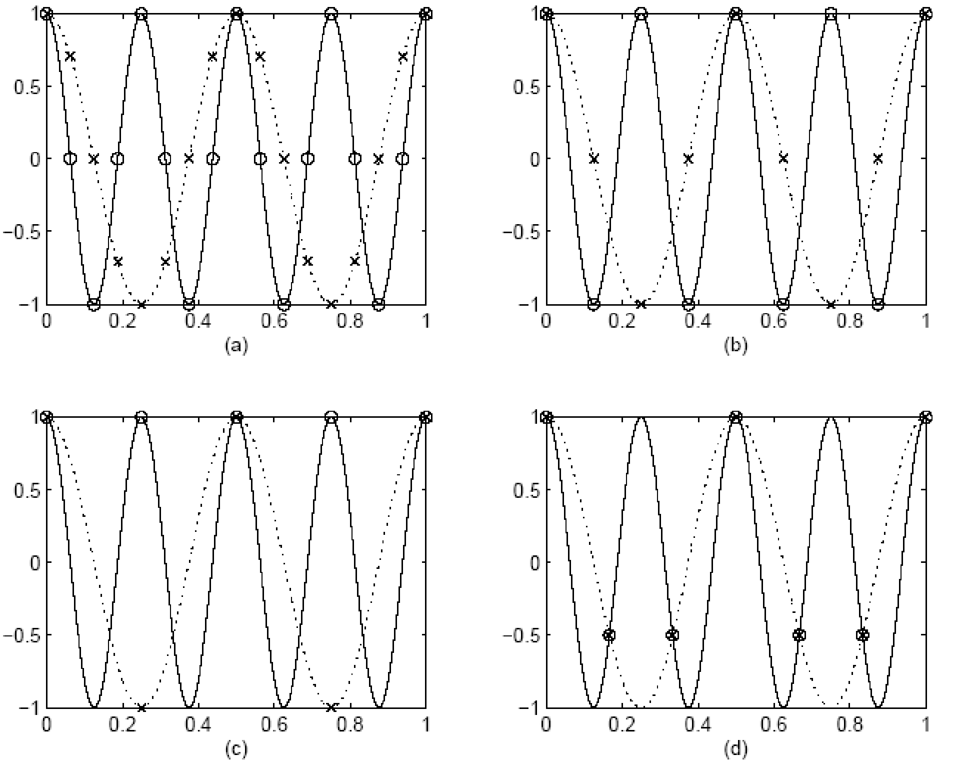


Figure 3.11: Two periodic signals $S(F_1)$ and $S(F_2)$ with respective frequency F_1 (dashed line) and F_2 (solid line) are depicted. Here, $F_2 = 2F_1$. The sampling points associated with the sampling frequency, F_s, are denoted by circles (○) and crosses (×). (a) $F_s = 4\ F_2$. (b) $F_s = 2\ F_2$. (c) $F_s = F_2$. (d) $F_s = 2/3F_2$. In both (a) and (b), the sub-sampled signal at F_2 retains its periodic structure and is distinct from F_1. In (c) and (d), the sub-sample signal at F_2 has lost it original periodic structure. Further, in (d) it is impossible to distinguish between F_1 and F_2 from the sub-sampled signal. Note that in (c), $S(F_1)$ is unaffected by aliasing since $F_s \geq 2F_1$.

Filters that cater for critical sampling, i.e., sampling at the *Nyquist* frequency [2], are particular useful since it ensure that the number of input sample prior to transformation equals the number of output samples after transformation. Any transform that has more output than input samples effectively expands data. While such *over-complete* transforms have been applied in pictorial compression, they are more challenging and complex to implement [37]. The *Nyquist frequency*, F_{Max}, is defined as the bandwidth (i.e., maximal frequency) of a signal [2]. The *Nyquist rate*, F_N, is defined as twice the *Nyquist* frequency. Sampling below the *Nyquist* rate results in signal aliasing [28] When aliasing occurs, signal components with frequencies, F_a, above half of the sampling frequency F_s, also known as the *folding frequency* F_f, are indistinguishable from those with frequencies, F_b, below the folding frequency F_f, such that $F_a = F_b + n \times F_f$, where n is an integer with $0 < F_b \leq F_f$ and $F_a > F_f$. This is illustrated in Figure 3.11 with time-domain sinusoidal signals.

For both auditory and pictorial signals, the appearance of aliasing typically results in high frequency signal distortions [2].

Aliasing in the transform domain occurs when down sampling is carried out on filters with overlapping frequency responses (Figure 3.12) [27, 29]. Typically, filters are designed with multiple overlapped response bands to cover the entire range of possible frequencies. Overlapping is a practical necessity since idealized filters[19], requiring infinite filter taps, is impossible to implement [29]. When these overlapped filters are down sampled, i.e., sampled at the critical frequency, aliasing is induced in the overlapped regions.

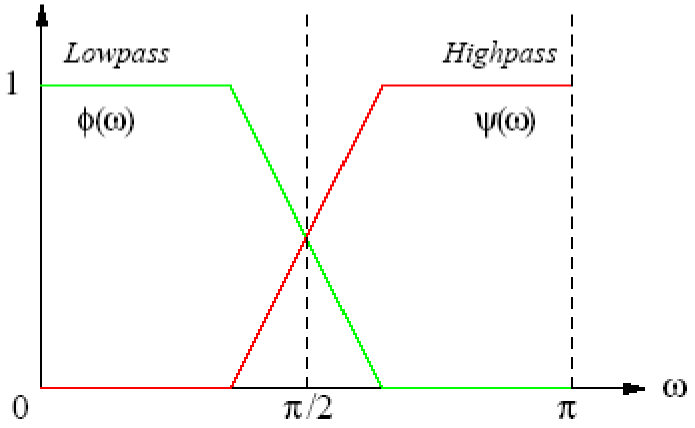


Figure 3.12: Overlapping frequency responses of a two band system with a low-pass ($\phi(\omega)$) and a high-pass ($\phi(\omega)$) filter. The overlapped region is the triangular area that centres around the mid frequency, $\pi / 2$.

Two classes of invertible filters which allow critical sampling and address the aliasing factor are the orthogonal[20] and bi-orthogonal filters. In these filters, aliasing is dealt with through alias cancellation in the reconstruction/synthesis process [27, 38]. Therefore, while aliasing still exists in transformed data, these aliasing components are effectively eliminated during the inverse transform. However, total alias cancellation is only possible if the transform data remains undamaged/unaltered. A condition that is impossible to satisfy if the quantization operation is undertaken after transform [27].

Differences between orthogonal and bi-orthogonal filters can been seen in structural terms. Orthogonal filters allow for *m* number of frequency response bands where as bi-orthogonal filters have dyadic response bands, one high pass and one low pass. In addition, orthonormal systems such as the DFT and DCT are inherently critically sampled and thus preserve vector length[21] [27]. Bi-orthogonal filters on the other hand do not preserve

[19] With perfect (non-overlapped) frequency localization.

[20] Orthogonal filters with unity gain are referred to orthonormal filters [27].

[21] The number of input samples equals the number of output samples

vector length. Consequently, they have more inputs than output. For image compression applications, this disparity may be resolved through the extension of input signal samples [15, 36]. In terms of filter properties, bi-orthogonal filters have a number of advantages over orthogonal filters such as phase linearity and regularity. These properties are discussed further in [27].

3.2.6.2 Quantization

In order to achieve an acceptable level of compression, it is necessary to limit the possible range of symbols to be coded after transformation. The quantization operation in image coders effectively re-maps the transform coefficient from a larger to a smaller set of discrete number. Quantizers may be scaler or vector based. In addition, scalar quantizers may be uniform or non-uniform (Figure 3.13). Scalar quantization is usually carried out through the division operation

$$X_q[n] = \left\lfloor \frac{X[n]}{q} \right\rfloor \tag{3.31}$$

where $X_q[n]$ is the scalar quantized symbol, $X[n]$ the transformed coefficient, q the quantizer and $\lfloor \ \rfloor$ denotes rounding down to the nearest integer. The methodology for designing optimal quantizers is discussed in [39, 40].

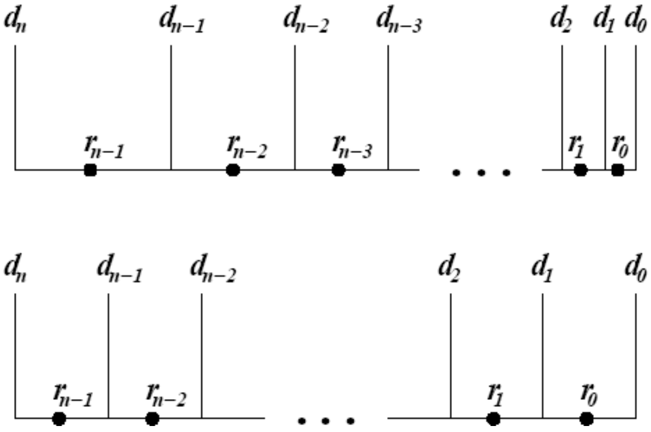


Figure 3.13: Non-uniform (above) and uniform (below) scalar quantization. Each set of decision levels, *d*, corresponds to a quantized response, *r*. For uniform quantization, the decision levels are equal distance apart. Non-uniform quantization has variable distance intervals between decision levels.

Quantization introduces quantization errors, *e*, defined as the difference between the de-quantized, $\overline{X}[n]$, and unquantized data sample:

$$e = X[n] - X_r[n] \tag{3.32}$$

where $X_r[n] = X_q[n] \times q$. Quantization error leads to the irrecoverable information loss in lossy coding. While scalar quantizer is quite effective in shaping a favourable PDS for entropy coding, it has been shown that *Vector Quantization* (VQ) is generally more effective than scalar quantization [41]. In fact, scalar quantization is seen a subset VQ where the vector length equals to unity. VQ operates by approximating a group of transformed coefficients[22], $\boldsymbol{X}[u] \in X$, from the transformed space, *X*, with a vector symbol, $V[k] \in V$. The vector symbol is chosen from a vector codebook, *V*, of size *M* based on some distance minimization criteria, f_{DMC}, such as the *Mean Squared Error* (MSE). With the MSE criterion, this process is formulated as

$$V[k] = f_{DMC}(\boldsymbol{X}[u], V) = \min(\mathrm{MSE}(\boldsymbol{X}[u], V[m])), \quad \forall m \in M\ . \tag{3.33}$$

The MSE is defined as

$$\mathrm{MSE}(\mathrm{A},\mathrm{B}) = \frac{1}{N}\sum_{n}^{N}(\mathrm{A}[n] - \mathrm{B}[n])^2\ . \tag{3.34}$$

The effectiveness of VQ is ultimately dependent of the size of the vector codebook, *M*. Having a large codebook with more vector codes would lead to better approximation. Unfortunately, as the size of the codebook grows, the overhead associated with storing and transmitting the codebook increases. Additionally, the computation for the vector matching operation also increases. Thus, there is a practical limitation that curtails the effectiveness of VQ.

Another realisation of quantization is recursive quantization[23] associated with progressive bitplane coding. This approach breaks the transformed coefficient into its constituent bit components and encodes each bit component progressively from the most significant bit (msb) to the least significant bit (lsb). The attraction of this method is that it is particularly suited for scalable coding.

3.2.6.3 Bitstream Coding

The final stage of a transform coder is concerned with two things. First, the entropy coding of quantized transformed data. Second, the efficient arrangement of entropy coded data stream. General details of entropy coding have been covered previously (section 3.2.3). For bitstream coding, it is common to adapt various combination of coding techniques to suit the nature of the data being coded. For example, the JPEG still image coder employs run-length [2] coding prior to Huffman coding in an effort to reduce the number of quantized coefficients to be coded. This is also true for standardized hybrid video coders

[22] current discussion centers on frequency domain transformed coefficients. However, VQ may also operates on time-domain data.

[23] Also referred to as progressive bitplane quantization.

[42]. Similarly, for the JPEG2000 coder, run-length coding has also been utilized, albeit conditionally. Further, the JPEG2000 coder employs context binary arithmetic coder for entropy coding to match the progressive bitplane quantization strategy. The formation of the final data stream arranges the order in which entropy coded data is stored. This is especially important where scalability is concerned. For example, if a picture is coded with resolution scalability in mind, then the final bitstream should be formed in order of picture resolution, i.e., from low resolution to high resolution.

3.2.6.4 Video and Image Sequence Coding

One approach to encoding digital video is to encode each video frame independently as seen in motion JPEG 2000 [43]. While this method does have some useful applications, particularly in video editing, its compression performance is unimpressive compared with three dimensional (3-D) transform coding [44-46, 81] and hybrid video coding techniques [10, 16]. Conventional hybrid video coding strategy (Figure 3.14) operates in two modes: *intra-frame* and *inter-frame*.

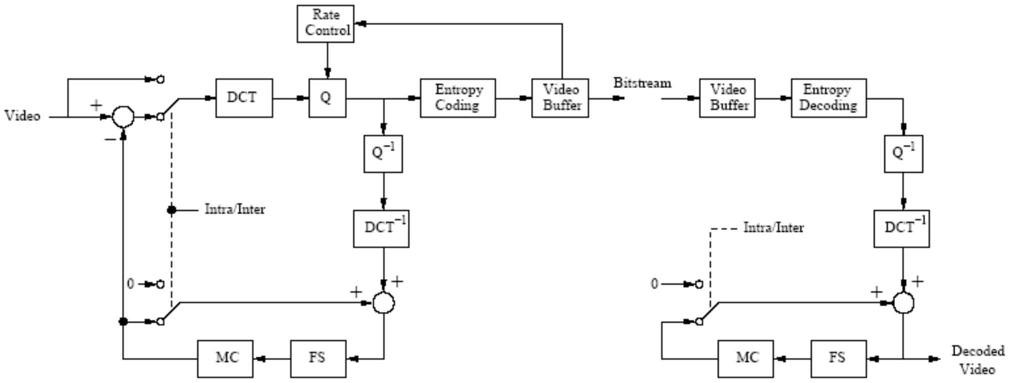


Figure 3.14: A simplified hybrid video coding structure (MPEG) that switches between intra and inter-frame coding. The difference between these two modes is the additional prediction operation for the inter-frame mode. In the intra-frame mode, DCT is performaed over video data followed by quantization (Q). The quantized data are then entropy coded to form the compressed bitstream. The entropy coding process also feeds into a rate control function which attempts to maintain coding at a desirable bitrate. For the inter-frame mode, the reconstruction of previously encoded frames is needed in order to generate the prediction for the current frame with motion compensation (MC). Consequently, a reverse process with de-quantization (Q^{-1}) and inverse DCT (DCT^{-1}) is encapsulated in the encoder. Frame storage (FS) is employed to store these previously encoded reference frames for MC operations. The decoding end mirrors the encoder, but in the reverse order.

The intra mode to all intents and purposes behaves as a still image coder and is intended for encoding *reference* frames only. The inter mode complements this by encoding the *difference* frames. While the physical encoding operation in the inter mode is equivalent to that of the intra mode, its primarily component is a prediction engine used to perform the *displace frame difference* (DFD) operation [10]. The DFD generates the difference frame based on the reference frame (Figure 3.15). The reference frame is then encoded in a similar manner to the intra-frame mode.

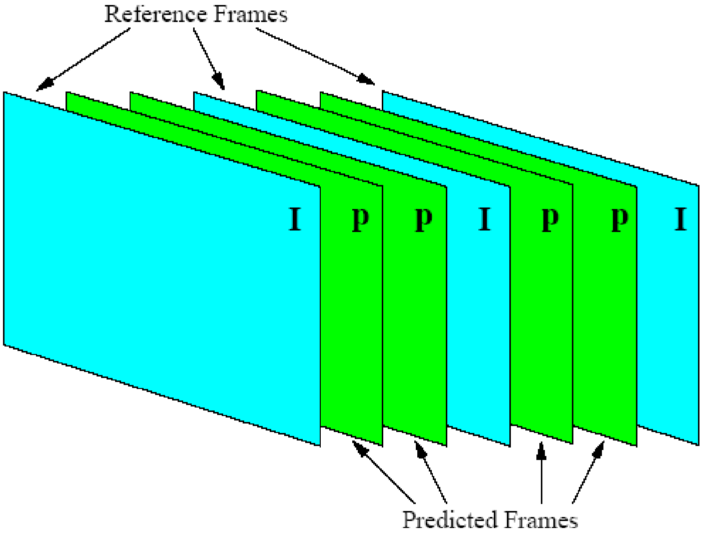


Figure 3.15: Reference (I) and difference (P) frames arrangement for hybrid video coding

In order to account for motion in video data for DFD operations, *Motion Estimation* (ME) and *Motion Compensation* (MC) [5, 7] functions are used to track moving objects in the temporal field. This has the effect of reducing difference or residue errors from DFD operations. However, even with MC and ME, motion mismatch may still occur resulting in high residue errors. 3-D transform coders have no standardized structure. Some of these coders are direct 3-D extension of their 2-D counterparts [47, 81]. Others, however, adopt the hybrid coding structure with ME/MC [44, 46, 48]. While 3-D transform coding is focused primarily on general digital video coding applications, it has also been proposed for coding of volumetric medical images [49].

3.2.6.5 Scalability

Scalability has been an issue of intense interest in picture coding in recent years. It is a flexibility feature that enables some degree of control over the variations between the encoding and decoding end with respect to picture quality, resolution and in the case of video, frame rate [10]. It is envisaged that scalability would be most useful in instances where pictures are coded to the optimum scale[24], but decoded at different scales, subject to the requirements and limitations at the decoding end [10]. An example of this may be a central picture repository, such as art collection or medical images, having high quality pictures which are accessed remotely. Scalability is also particularly useful in transcoding where digital videos are re-encoded to different bitrates, quality or resolution [50]. To

[24] Scale in reference to quality, rate and resolution.

maintain scalability in picture coding, it is necessary to arrange the bitstream of the coder into appropriate layers that reflects the scale used. For example, if pictures are scaled according to quality, then the layers should be arranged according to layers of decreasing degree of quality improvement [51, 52]. That is, the first layer should have the highest degree of quality improvement and the last layer should have the least.

3.2.7 Perceptual Picture Coding

The impact on picture quality in lossy coding has always been an area of concern, particularly for high quality images. Traditional methods for quantifying distortions and picture quality do not consider the human factor [53]. It is recognized that picture quality is dependent on perceived picture content. As a result, it has become increasingly common for picture coders to incorporate, at the very least, some aspects of the *Human Vision System* (HVS) into the coding system. HVS based coding may operate at two levels, above or below threshold vision. The former being perceptually lossy while the latter is perceptually lossless. The conceptual view of perceptual coding and traditional lossless and lossy coding is depicted in Figure 3.16.

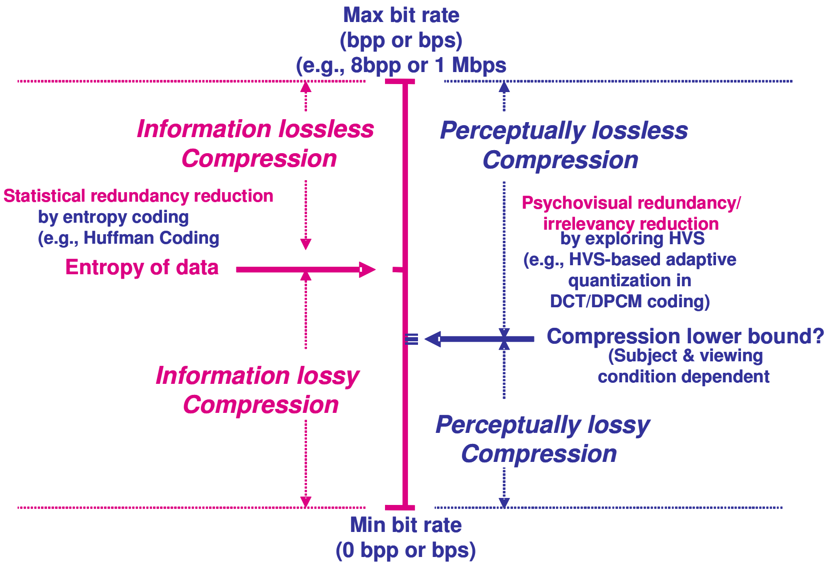

Max bit rate
(bpp or bps)
(e.g., 8bpp or 1 Mbps
Information lossless Compression
Statistical redundancy reduction by entropy coding (e.g., Huffman Coding
Entropy of data
Information lossy Compression
Perceptually lossless Compression
Psychovisual redundancy/ irrelevancy reduction by exploring HVS (e.g., HVS-based adaptive quantization in DCT/DPCM coding)
Compression lower bound? (Subject & viewing condition dependent
Perceptually lossy Compression
Min bit rate
(0 bpp or bps)

Figure 3.16: Conceptual view of picture coding philosophy from [54].

There are two main issues in perceptual coding. The first is the modelling of the HVS for which a detailed treatment is given in [55]. Human vision has a physical and a psychological aspect. The psychological aspect is concerned with the human mind, specifically, what the mind itself perceives based on memories and experiences. The physical aspect deals the physiology of the HVS which broadly consists of the human eye, the visual pathways and the visual cortex. The level of understanding between these three physical components varies. For example, literary knowledge of the human eye is sufficiently well developed to allow accurate modelling of the visual acuity of the eyes and colour sensitivity. However, the more exact mechanical operation of the visual pathways and, specifically, the visual cortex is less developed. Hence, the model of the visual cortex only approximates basic primitive functions of the visual cortex [56]. Nevertheless, this model does offer a credible behavioural approximation of the HVS. The basic physiological characteristics of the HVS can be surmised as follows:

- It is sensitive to the frequency and orientation of visual stimuli;
- It operates in a relative rather than an absolute manner. More specifically, the HVS sees the contrast between the luminance of two visual stimuli rather than the absolute luminance level of the two stimuli.
- The visibility of visual stimuli may be affected by masking and facilitation. Masked signals would have diminished visibility while facilitated signals increase their visibility.

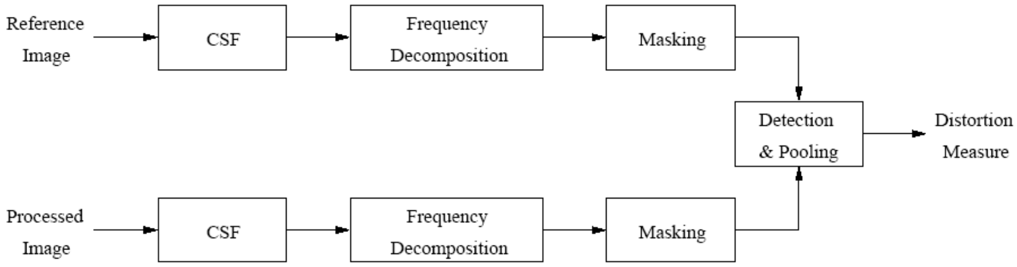


Figure 3.17: The Contrast Gain Control Model.

The *Contrast Gain Control* (CGC) model [57] (Figure 3.17) provides a reliable generic description of the HVS. Functionally, it evaluates the perceived similarity/difference between two images, a reference and an altered copy of the reference. The CGC model consists of four main parts: contrast estimation, filtering, masking and pooling. Contrast estimation translates images from the absolute scale to the relative scale via the *contrast sensitivity function* (CSF). This is followed by the filtering operation which projects images from the time domain into an oriented frequency domain. There are a number of realizations for the filtering operation including the Steerable Pyramid Transform [58], Cortex Transform [59] and Gabor Array [57] to name a few. Masking then attenuates individual coefficient sample, in the oriented frequency domain, according to the activity of their surroundings. The masking response, R_m, has the general form

$$R_m[l,\theta,i,j] = \upsilon_{m,1} \cdot \frac{X[l,\theta,i,j]^{p_m}}{h_m^{q_m}[l,\theta,i,j] + \upsilon_{m,2}} \tag{3.35}$$

where the transform coefficient $X[l,\theta,i,j]$ is masked by an inhibition factor $h_m^{q_m}[l,\theta,i,j]$ of masking domain, *m*. Variables *l*, θ and (*i*, *j*) specifies the frequency, orientation and spatial location of the transform coefficient, respectively. p_m, q_m, $\upsilon_{m,1}$

and $\upsilon_{m,2}$ are parameters to be optimized. Masking may occur in spatial, frequency and/or orientation domain [57]. Note that (3.35) provides a separable model for quantifying masking in individual domain. An alternative model that concurrently quantifies masking in all domains has been proposed in [57]. Typically, only spatial and/or orientation masking are considered [51, 60, 61]. For spatial masking, the inhibition factor is a measurement of the activity surrounding the target coefficient, $X[l,\theta,i,j]$, given by

$$h_s^{q_s}[l,\theta,i,j] = \sum_u^U \sum_v^V X[l,\theta,u,v]^{q_s} \tag{3.36}$$

Spatial masking is localized within a windowed area as specified by *U* and *V*. This window area is generally centred on spatial location (*i*, *j*). Orientation masking measures the activity over same spatial location, but over orientation, ϕ. It is written as

$$h_o^{q_o}[l,\theta,i,j] = \sum_\phi^\Phi X[l,\phi,i,j]^{q_o} \tag{3.37}$$

Separable masking models may be unified into a single measurable quantity by weighted summation during pooling as in [60]. The pooling stage sums all the differences between masking responses of the reference and processed image. The Minkowski summation is generally used for this purpose. It is defined as

$$D_{CGC} = g_m \cdot \sum_m^M \left(\sum_l^L \sum_\theta^\Theta \sum_i^I \sum_j^J \left(R_m[l,\theta,i,j] - \overline{R}_m[l,\theta,i,j] \right)^\beta \right)^{\frac{1}{\beta}} \tag{3.38}$$

where g_m is the weight for masking domain *m*. $R_m[l,\theta,i,j]$ and $\overline{R}_m[l,\theta,i,j]$ are the masking response of the reference and processed pictures, respectively. The overall distortion spans all resolution levels (*L*), orientations (Θ), and spatial locations (*I*,*J*).

The second issue regarding perceptual coding is the adaptation of the HVS models to picture coders. There are a number of ways to adapt vision models to coding structures. The most common method is through the quantization stage where vision models regulate the quantization operation to control the level of perceived distortions in coded pictures [62-64]. An unusual method to adapting HVS to coders is to design transformation filters that are specifically tuned to aspects of the HVS [65]. For scalable coders such as the JPEG2000 still image coder, vision models may be adapted to the error distortion measure in the *Rate-Distortion* (R-D) function [51, 60, 61].

Perceptual coders may be rate-driven or quality-driven in nature. The purpose of the rate-driven perceptual coder is to encode pictures to the best possible visual quality for a given bitrate. In the quality-driven coder, pictures are encoded to a desired visual quality level at the lowest possible bitrate. A quality-driven coder operating at just below the super-threshold level would provide visually lossless quality coding. The super-threshold level is defined as the point at which differences between two visual stimuli are just perceptible [55]. It is also commonly referred to as the *Just-Noticeable-Difference* (JND) level.

3.2.8 Standardized Coders

Under the auspices of the *International Standards Organization* (ISO), the *Joint Photographic Experts Group* (JPEG) and the *Moving Pictures Experts Group* (MPEG) have been the primary entities responsible for the development of industry standard picture coders. JPEG is responsible for still image compression that includes grey level and colour images for both lossy and lossless encoding. The JPEG-LS coder is a lossless coder based on the LOCO coding engine (see section 3.2.5.2). The LOCO coder was a fairly advance coding system when it was first introduced. In the years since, more efficient lossless coding algorithms have appeared [66, 67].

The JPEG baseline is a lossy image coding utilizing the DCT, scalar quantization with run-length and Huffman codes for entropy coding [68]. It is considered to be superseded by the recent state-of-the-art JPEG2000 coder. The JPEG2000 still image coder [36] employs *discrete wavelet transform* (DWT) with progressive bitplane coding. Entropy coding is handled by a moderately sophisticated context based arithmetic coder. This coding system is scalable in resolution and compression rate. In addition, it could operate in both lossless and lossy mode. While its performance in the lossy mode is superior the JPEG baseline, its performance in the lossless mode is inferior to the JPEG-LS [69].

MPEG is responsible for video compression technology. Its MPEG-x series of coders is closely related to the *International Telecommunication Union* (ITU) line of H.26x coders. The most recent standardized video coder, the MPEG-4 AVC/H.264 [16], has been developed under joint efforts between ISO and ITU. Both the MPEG-x and H.26x coders share a similar hybrid DCT/MC coding framework. Over the past decade, the underlying hybrid technology has matured considerably and substantial performance gains have been made in MC, entropy coding and post-filtering. The performance of MPEG-4 AVC/H.264 over its predecessors is analogous to the performance of JPEG2000 over JPEG baseline [70].

3.2.9 Applications of Picture Coding to Medical Images

Picture compression has become increasingly important to medical imaging. The shift toward the digital media has provided more flexibility in the way by which medical images are taken, transported and made available for diagnosis, e.g., Telemedicine [82]. As in many other applications, digitized medical images require storage space and bandwidth for transportation over communication networks. The problem arises when storage and transmission requirement exceed capacity. While it is possible to increase capacity, it may also be prudent to invest some effort in compression so as to reduce the storage and transmission requirements.

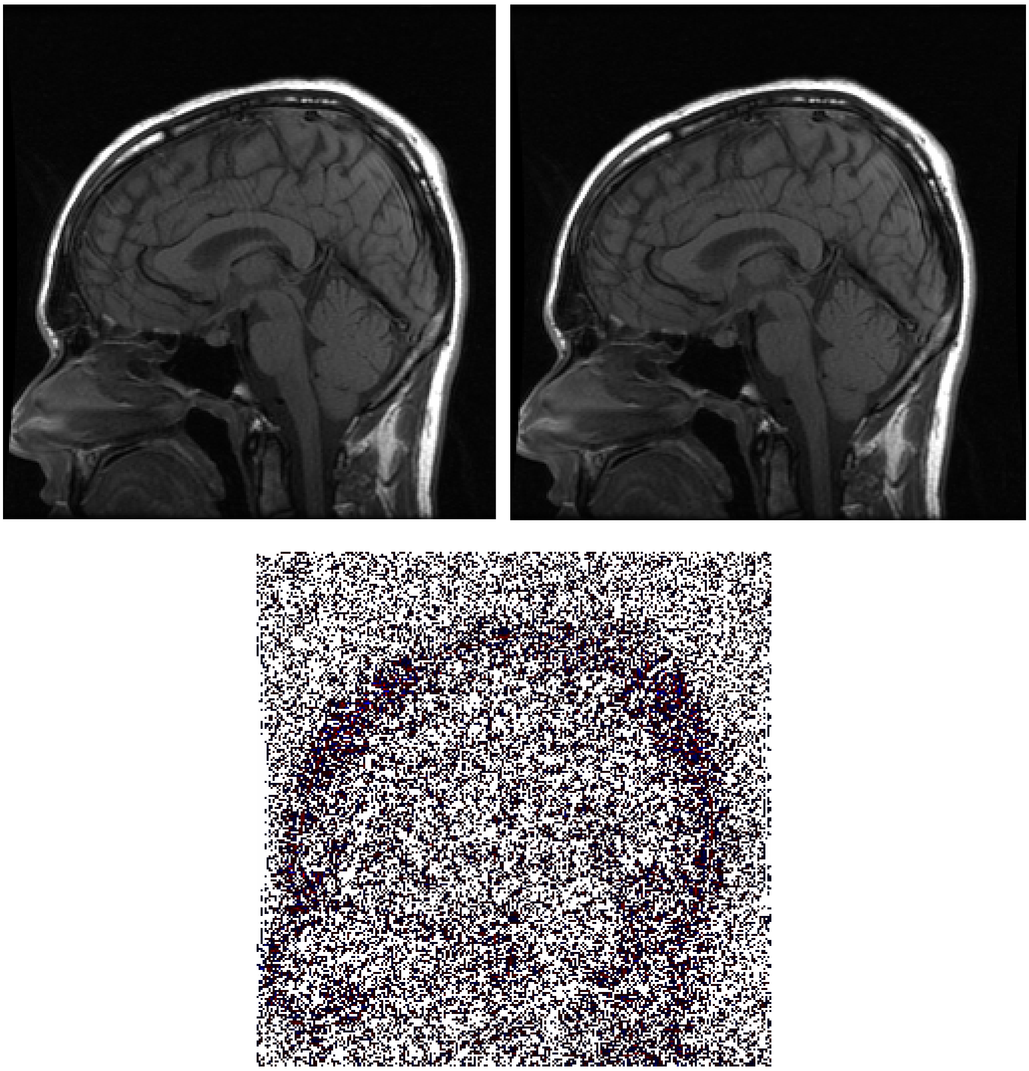


Figure 3.18: MRI slice of a brain. *Top left*: Original image. *Top Right*: Perceptually lossless coded image [71]. *Bottom*: Difference image between original and perceptually lossless coded image. For the difference image, white areas indicate no pixel difference while dark areas contain some pixel difference. The difference image reveals areas within the image where psychovisual redundancies exist.

The critical issue for medical images in regards to compression is information integrity. Information loss should be avoided when possible. Thus, in this regard, the medical fraternity sees compression as lossy or lossless. In situation where information loss is unavoidable due to practical reasons, then the attention is shifted from prevention to minimization of information degradation [72]. The deterioration of information through compression may be deemed acceptable if the diagnostic value of medical images is preserved. The question for lossy medical coding then is two folds. First, under what conditions could lossy compression be used? Second, what is the error tolerance level for maintaining the diagnostic value of medical images? For example, which pixels in a medical image contain critical diagnostic information? Moreover, what effects do distortions from lossy coding have on diagnostic quality? Currently, there are no formal guidelines for use of lossy coding in medical images. This may be partly due to legal considerations. The possibility of the loss of some diagnostic information may lead to a drastic misdiagnosis has considerable legal ramifications. Until lossy compression could guarantee the preservation of diagnostic information in medical images, it is likely that medical imaging will focus more on lossless compression [73].

If the diagnostic value of medical images is taken as measurable perceptible quantities, then perceptual coding may be a solution that clinically retains diagnostics information of medical images. To this effect, perceptual lossless coding has been shown to be equally or more effective than the lossless and the near-lossless coding strategies [71]. Ultimately however, the compression of medical images may be dependent on the nature of the diagnosis and individual situation.

3.3 COMPRESSION IN THE DICOM STANDARD

This section will provide a brief description of picture coders supported under the image compression component of the *Digital Imaging and Communications in Medicine* (DICOM) standard Chapters 2 and 13 of this book and [74]. The DICOM standard provides a standardized format for collating all information associated individual medical images. It encapsulates pictures compressed through standardized coder within its structure, thus ensuring modularity. This modular arrangement allows for future introduction and retirement of coders to and from the standard. Once medical images are bound to the DICOM format, the manner in which they are stored and transmitted is covered by the *Picture Archiving and Communication System* (PACS) Chapter 13 and [75, 76].

3.3.1 DICOM Recommended Coders

DICOM does not necessarily support all features of standardized coders. Additionally, DICOM neither specify nor recommends under what conditions lossy compression should be used. The decision is left entirely to individual users. Coders that are currently supported in the DICOM standard are:

- JPEG-LS [22] for lossless and near-lossless compression based on the LOCO coder [20].
- JPEG baseline [68, 77] for lossy compression. Implements DCT with scalar quantization and Huffman coding.
- JPEG2000 [15, 36] that supports both lossless and lossy compression through reversible (5/3) and irreversible (9/7) filters, respectively. It also supports scalable coding. Utilize wavelet transform with bitplane coding and arithmetic coding.

- MPEG-2 MP@ML (main profile at main level) [42] for compressing multi-frame images.

3.3.2 Image Modality

Diagnostic imaging falls in three general categories: transmission, reflection and emission imaging. Transmission imaging such as roentgenography (x-ray) projects particles through a medium to capture specific characteristics within that medium. The acoustic based reflection techniques send pulse signals into a medium. Information within the medium is then captured from reflected signal pulses. Emission imaging operates by capturing signal emanations from within a medium. These emanations may be induced externally, through the injection of radioactive isotopes, as in the case on nuclear medicine or come about naturally in magnetic resonance imaging (MRI).

The modality of a medical image specifies the method by which the image is captured, e.g., Magnetic Resonance (MR), Computed Tomography (CT) and ultrasound, etc. Different modalities are intended to extract different types of information. A list of supported modality is provided in Chapter 1 and [74]. The DICOM standard provides no recommendation as to what type of compression, lossy or lossless, should be used for any particular modality. This decision is left to individual user. One factor that may affect the choice of compression system is the size of digitized images. Medical images have the tendency of being sizable due to their bit depth, that ranges between 8-bit and 16-bit, and have resolution that may exceed 1000×1000 pixels [78]. In addition, when dealing multi-frame/sliced images, the amount of storage requirement becomes most noticeable. Therefore, it may be more economical to have high compression with some information loss if the diagnostic value of medical images can be maintained [79, 80].

3.4 DATA COMPRESSION FOR DYNAMIC FUNCTIONAL IMAGES

This section will present the medical data compression for multidimensional dynamic functional images based on various diagnostically lossless coding schemes. A brief background on multidimensional dynamic functional imaging studies with physiological parameter estimation is firstly given, followed by addressing the need for developing efficient multidimensional data compression to reduce the volume of dynamic functional images without affecting physiological parametric estimation and clinical decision making. After that, diagnostically lossless compression techniques for dynamic functional image data are described in three aspects, i.e., *compression in temporal domain* based on an optimal image sampling schedule, *compression in spatial domain* with clustering analysis, and *compression in sinogram domain* by a combination of principal component analysis and a channel-weighted JPEG2000 coding scheme.

3.4.1 Multidimensional Dynamic Functional Imaging Studies

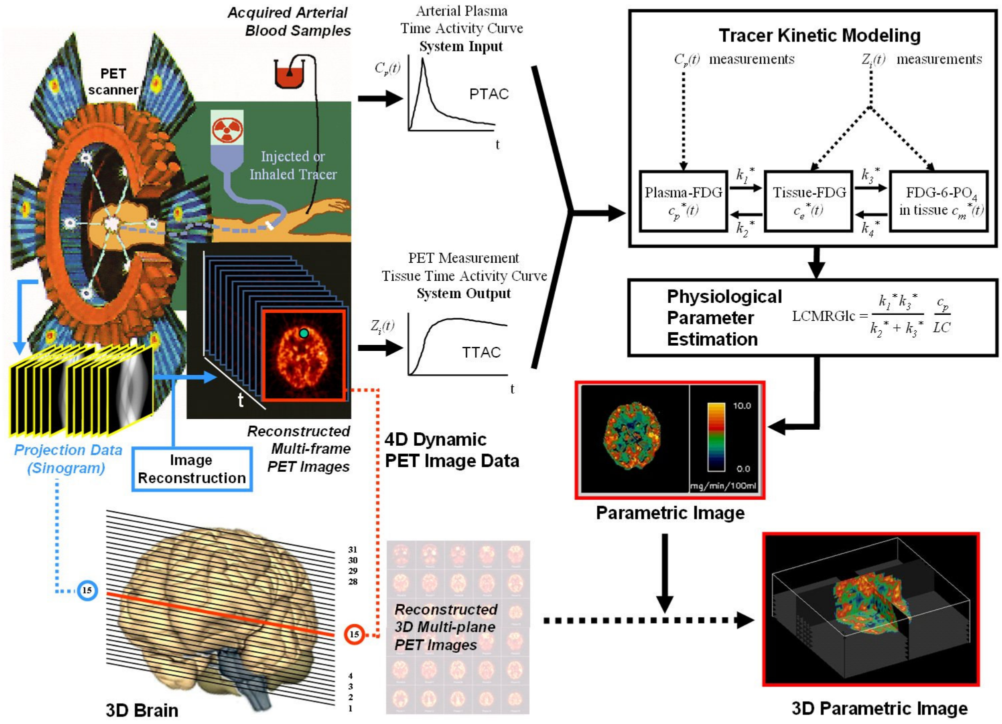

Acquired Arterial Blood Samples
PET scanner
Injected or Inhaled Tracer
Arterial Plasma Time Activity Curve System Input
$C_p(t)$
PTAC
t
PET Measurement Tissue Time Activity Curve System Output
$Z_i(t)$
TTAC
t
Projection Data (Sinogram)
Image Reconstruction
Reconstructed Multi-frame PET Images
4D Dynamic PET Image Data
Tracer Kinetic Modeling
$C_p(t)$ measurements
$Z_i(t)$ measurements
Plasma-FDG $c_p^*(t)$
Tissue-FDG $c_e^*(t)$
FDG-6-PO$_4$ in tissue $c_m^*(t)$
k_1^*
k_2^*
k_3^*
k_4^*
Physiological Parameter Estimation
$\mathrm{LCMRGlc} = \frac{k_1^* k_3^*}{k_2^* + k_3^*} \frac{c_p}{LC}$
10.0
0.0
mg/min/100ml
Parametric Image
31 30 29 28
15
4 3 2 1
3D Brain
Reconstructed 3D Multi-plane PET Images
3D Parametric Image

Figure 3.19: A brief diagram for the process of generation of physiological parametric images based on tracer kinetic modeling – for example, quantitative estimation of regional glucose metabolic rate with FDG PET. After intra-venous injection of the FDG tracer, the time course of the regional radio-tracer concentration in the brain is obtained by acquiring a series of images. At the same time, the input function is obtained from a series of blood samples. The physiological parameter of interest, in this case the local cerebral metabolic rate of glucose, is estimated by fitting a compartmental model to the data. Four-dimensional data (three dimensions in space and one in time) are required to construct the three-dimensional parametric image, which depicts regional glucose metabolism quantitatively in quantitative units of mg/100g/min.

As mentioned in Section 1.4, dynamic functional imaging such as positron emission tomography (PET) in nuclear medicine can provide image-wide quantification of physiological, pharmacological and biochemical functions within the body, and support the visualization of the distribution of these functions corresponding to anatomical structures. Physiological function can be estimated by observing the behaviour of a small quantity of an administered substance ‘tagged’ with radioactive isotopes (tracers). Images are formed by the external detection of gamma rays emitted from the patient when the tracers decay. Since they allow the observation of the effects of physiological processes, where most diseases are functional in nature and structural changes are secondary, functional imaging techniques are invaluable aids to patient diagnosis and treatment [83]. The range of tests that can be performed in functional imaging studies is extremely large and covers all organ systems of the body. In some studies, the time course of tracer redistribution from the time of administration onward must be observed and quantified to enable the calculation of physiological parameters by tracer kinetic modelling. Figure 3.19 illustrates a typical study for processing and analysis of dynamic functional image data, and subsequent generation of human brain parametric images using PET with the glucose tracer ^{18}*F-fluoro-deoxyglucose* (FDG). To estimate physiological parameters and form parametric images, for each cross-section plane, the PET scanner acquires a series of scans at a pre-determined rate (not necessarily constant), typically for 20-60 minutes, in which projection views (sinogram data) are acquired at multiple angles, and reconstructed into slices which depict regional tracer uptake and function during the study. From these data a tissue time-activity curve (TTAC) can be plotted for each voxel, and the physiological parameter value for that voxel calculated by the application of a tracer kinetic model to the TTAC. If the modelling process is repeated for every plane, then a 3D physiological parametric image can be constructed [84]. More details of generating parametric images can be found in Chapter 6.

The above dynamic functional imaging studies, however, are accompanied by a growth in the size of the image data. For example, a routine dynamic PET study using a CTI 951 scanner (CTI Inc., Knoxville, TN) typically involves the acquisition of 31 cross-sectional image planes of 128×128 pixels each, at 20 to 30 time points. The resulting four-dimensional data set contains upwards of 11 million data points, requiring approximately 22 megabytes of storage for just one study for one bed position. Such a large number of images place a considerable load on computer storage space and retrieval, data processing and transmission time. When the image resolution has to be improved or the scanning procedure has to include the whole body, rather than just a single organ, the demand for space is greatly increased. Therefore, techniques for dynamic functional image data compression are of great interest. As mentioned in Section 3.2.4, conventional image compression algorithms can be divided into two major categories, lossless and lossy compression algorithms. Lossless compression algorithms allow for perfect reconstruction of the original images from compressed data. These algorithms yield modest compression ratios, typically between 1.7:1 to 2.1:1 for medical image data. On the other hand, lossy compression can achieve higher compression ratios. However, the original images can only be reconstructed approximately from compressed data, though the differences may not be distinguishable by the human visual system. The challenge in the development of a practical image compression scheme for dynamic medical images is the development of compression algorithms that are lossless for diagnostic purposes, i.e., make no difference to doctors qualitative and quantitative assessment, yet attain high compression ratios to reduce storage, transmission, and processing burdens. It should be noted that in the clinical situation a slight loss of precision in a derived parameter may be undetectable visually, and quite insignificant relative to the measurement error. The conventional compression algorithms are not specifically tailored for the diagnostic use of dynamic functional image data. Therefore, new algorithms have to be developed to fully exploit spatial and temporal redundancies in these data. In addition, the variation of data can be organized in such a novel way to remove the measurement noise and improve the measurement reliability. In the following sub-sections, three different diagnostically lossless compression techniques for dynamic functional image data will be reviewed, i.e., compression in temporal domain based on an optimal image sampling schedule [85-87], compression in spatial domain with clustering analysis [86-88], and compression in sinogram domain by principal component analysis and a channel-weighted JPEG2000 scheme [89-91]. For simplicity and clarity, the dynamic functional image data in human brain FDG-PET studies are used to illustrate the practicality of the above compression techniques.

3.4.2 Diagnostically Lossless Compression in Temporal Domain

In dynamic functional imaging studies, the reliability of the temporal frames is directly influenced by the sampling schedule and the duration of each frame. The number of counts and hence the statistical reliability of a frame increases with its duration. However, in order to obtain quantitative information about dynamic processes, a certain number of temporal frames are required. Conventional sampling schedules [92-95] which involve the acquisition of a large number of temporal frame images have been empirically developed and may not be optimal for the extraction of accurate physiological parameter estimates. Most previous studies suggest that a higher sampling frequency should be used over the early stages. This conclusion, however, imposes a considerable burden on the computer image storage space and data processing. To remedy the above limitations, an optimal image sampling schedule (OISS) has been developed and it has been demonstrated that the use of OISS is an effective way to reduce image storage requirements while providing comparable parameter estimates [85, 96]. It was found that if a different cost function for parameter estimation is used, which depends only on the direct PET measurement, rather than the instantaneous measurement, the accuracy of parameter estimation can remain almost unchanged when two neighbouring image frames are combined into one.

Finding the optimal image sampling schedule involves minimising the determinant of the convariance matrix of the estimated parameters **p**, or conversely maximising the determinant of the *Fisher Information Matrix* [97] by rearranging the sample intervals, using the minimum number of required samples. To illustrate the practicality of the OISS algorithm, the five-parameter FDG model [98] for describing the behaviour of FDG in brain tissue with the effects of cerebral blood volume is adopted in this section, in which the first four parameters $k_1^* \sim k_4^*$ represent model transport and reaction rate constants and the fifth parameter *CBV* is used to depict the effects of cerebral blood volume. For the five-parameter FDG model with:

$$\mathbf{p} = \lfloor k_1^*, k_2^*, k_3^*, k_4^*, CBV \rfloor, \tag{3.39}$$

the information matrix **M** with elements m_{ij}, can be represented as:

$$\mathbf{M} = [m_{ij}] = \left[\sum_{k=1}^{N} \frac{1}{\delta^2(t_k)} \left(\frac{\partial C_i^*(t_k, \mathbf{p})}{\partial p_i} \right) \left(\frac{\partial C_i^*(t_k, \mathbf{p})}{\partial p_j} \right) \right] \tag{3.40}$$

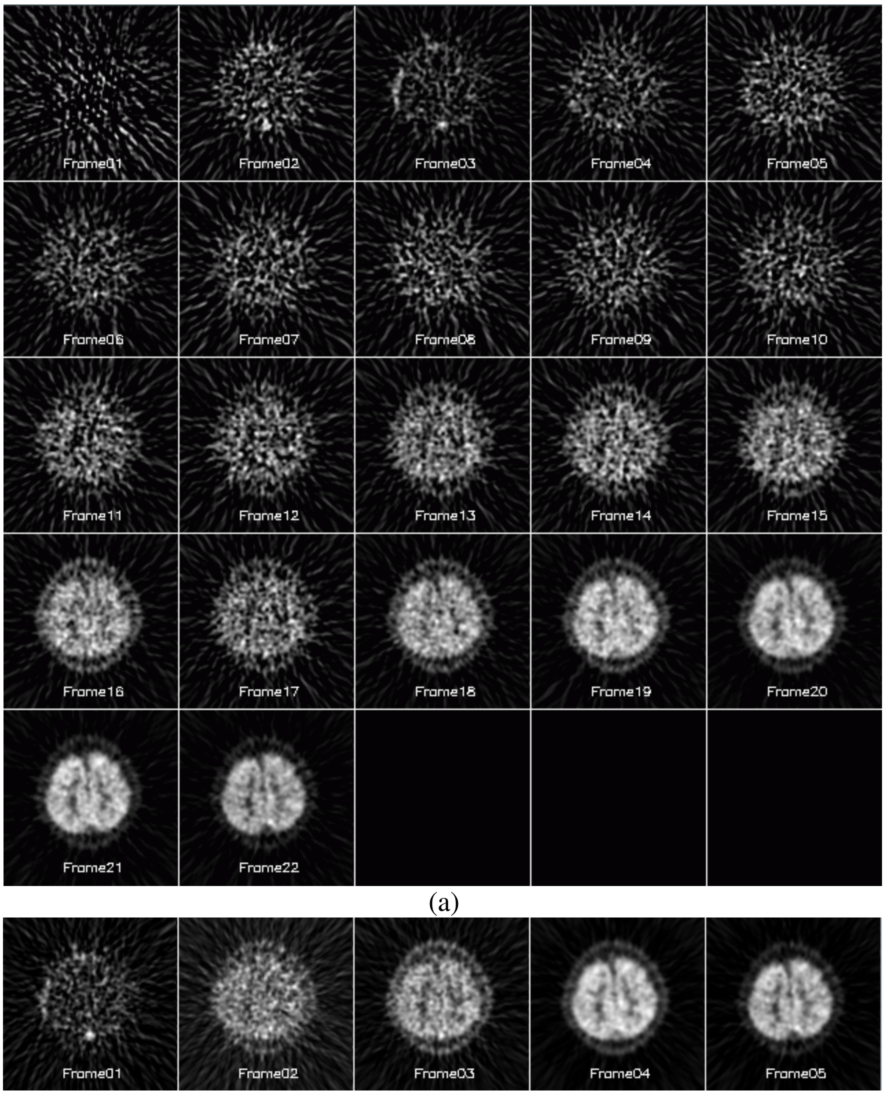
here $C_i^*(t_k, \mathbf{p})$ is the output function of the five-parameter FDG model:

$$C_i^*(t) = \frac{k_1^*(1-CBV)}{\alpha_2 - \alpha_1}\left\{(k_3^* + k_4^* - \alpha_1)e^{-\alpha_1 t} + (\alpha_2 - k_3^* - k_4^*)e^{-\alpha_2 t}\right\} \otimes C_p^*(t) + CBV \cdot C_p^*(t) \tag{3.41}$$

where $\otimes$ is the convolution operator, $C_p^*(t)$ is the FDG concentration in plasma represented by the plasma time activity curve (PTAC), and

$$\alpha_{1,2} = \left(k_2^* + k_3^* + k_4^* \mp \sqrt{(k_2^* + k_3^* + k_4^*)^2 - 4k_2^* k_4^*}\right)\Big/2 \tag{3.42}$$

The required sampling schedule can be adjusted iteratively to maximise the determinant of **M**, det(**M**), using an automatic algorithm [85]. A set of *a priori* parameters has to be provided as the nominal parameters of the model. The optimization procedure starts with an initial sampling schedule $\{I_1, I_2, \ldots, I_N\}$ and then iteratively adjusts the sample intervals. At each iteration, each interval is inspected and adjusted towards the direction in which det(**M**) increases. The optimization procedure always converges, as there are a finite number of intervals to start with and det(**M**) increases monotonically. It has been shown [85, 99] that the minimum number of temporal frames required is equal to the number of model parameters to be estimated. Therefore, for the five-parameter FDG model, five temporal frames should be sufficient to obtain parameter estimates with similar statistical accuracy and reliability to the conventional technique which typically requires the acquisition of more than twenty temporal frames. This reduces the number of temporal frames obtained and, consequently, reduces data storage. Furthermore, as fewer temporal frames are reconstructed, the computational burden posed by image reconstruction is reduced. Figure 3.20(a) shows the original 22 temporal frames for the 15th plane from one patient study. Due to the lower tracer concentration and short acquisition time in the first few frames, these images have been scaled to be visible. A set of 5 temporal frame images derived from the OISS algorithm is illustrated in Figure 3.20(b), where a compression ratio (CR) of 4.4:1 is obtained.



(a)

(b)

Figure 3.20: A set of 22 temporal frame images for the 15th cross-section plane from a patient brain FDG-PET study; (b) Results of compressing the images in the temporal domain with the OISS algorithm [87]

3.4.3 Diagnostically Lossless Compression in Spatial Domain

The data compression introduced in Section 3.4.2 is mainly used for exploiting temporal redundancies in dynamic functional image data. In terms of removing spatial redundancies in the data, for each cross-section plane, the reduced set of temporal image frames can be further compressed to a single indexed image using cluster analysis. In general, a tissue time-activity curve can be obtained from each voxel in dynamic PET images. However, many TTAC curves have similar kinetics. Clustering techniques therefore, can be adopted to automatically classify pixels into a certain number of typical TTAC types corresponding to different brain regions. The main idea behind clustering algorithms is to group and classify image-wide TTACs, $z_i(t)$ (where i=1,2,…,R, and R is the total number of image voxels), into S cluster groups C_j (where j=1,2,…,S, and S<<R) by measurement of the magnitude of natural association (similarity characteristics). In [86], an indirect agglomerative clustering algorithm was used to conduct the data compression in spatial domain, based on the traditional Euclidean distance criterion measure:

$$D^2\left(z_i, \bar{z}_{C_j}\right) = \sum_t \left[z_i(t) - \bar{z}_{c_j}(t)\right]^2 \tag{3.43}$$

where $\bar{z}_{c_j}(t)$ denotes the mean TTAC within each cluster C_j. Here the clustering analysis technique is applied to further compress the reduced set of temporal frames into (1) a single indexed image which represents a mapping of the cluster groups to their respective pixel TAC locations (i.e. the spatial distribution of kinetic behaviour), and (2) an index table which contains the mean TAC for each cluster group. In contrast with other kinds of medical images, dynamic PET images have a consistent general structure consisting of an approximately oval region containing almost all of the information of interest. Therefore, pixels containing background noise and negative values were suppressed prior to cluster analysis in order to get the accurate clustered TTAC results. Using cluster analysis, the reconstructed images were further compressed in the spatial domain, and a compression ratio of 8.6:1 can be gained. Furthermore, PNG (Portable Network Graphics) [100], a well-known standard image lossless compression method, can be used to compress the single indexed image, achieving a further compression ratio of 1.8:1. The PNG coding format was chosen over other lossless image compression methods due to its efficiency, portability, flexibility and it being legally unencumbered. In many centers performing clinical dynamic PET studies, the extraction of physiological parameters, i.e. the generation of parametric images, is of major importance. With the proposed compression technique, it can be implemented through (1) de-compression of indexed image; (2) tracer kinetic modeling and parameter estimation; and (3) pixel-wise mapping. The resultant images obtained from the compressed data correspond to the generated functional images [86]. Figure 3.21 shows the result of applying cluster analysis to the temporal image frames in Figure 3.20(b). Compression does not appear to have degraded image quality and fine structural information of human brain, while the overall compression ratio obtained for the combined compression approach in temporal and spatial domains is 68.1:1 [87].

In the above cluster analysis algorithm, the number of clusters used in dynamic functional image data is a critical issue. A sufficient number of clusters are usually required to ensure that the functional data contained in the dynamic images is adequately represented, however, too many clusters will increasingly reflect the variation in the TTACs due to noise and will increase the heterogeneity in the index image, resulting in increased noise and less scope for compression of the index image. A performance evaluation for compression of dynamic brain FDG-PET image data has been conducted in [88], and the optimal number of clusters was shown to be 21~42. For a cluster number of 42, the compression ratio achieved was approximately 87:1, while the minimal practical number of clusters was shown to be 21 and this gives a maximum compression ratio of approximately 86:1 [88].

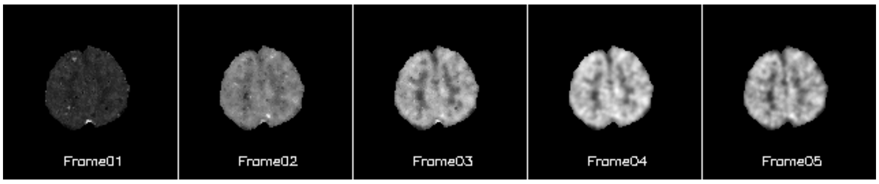


Figure 3.21: Results of applying cluster analysis to the temporal image sequence in Figure 3.20(b) [87]

3.4.4 Diagnostically Lossless Compression in Sinogram Domain

Dynamic functional imaging studies using a conventional sampling schedule (CSS) produce large numbers of temporal image frames which may not provide the maximum information for the study. In Section 3.4.2, the OISS has been shown to greatly compress dynamic functional image data in temporal domain – reducing the number of time samples required to the number of model parameters that are being estimated while providing good parameter estimates. However, the OISS is model-dependent and requires an input function, typically from arterial blood sampling, which is complicated to retrieve. Moreover, the OISS is optimised for a specific model rather than each individual patient dataset, and conventional compartment models may not be adequate to describe heterogenous tumour tissues that require more complex modelling [101, 102]. An alternative model-independent approach for dimensionality reduction of dynamic PET data is multivariate data analysis techniques such as principal component analysis (PCA) [103]. Previous PCA-based temporal compression approaches for dynamic PET data have been applied in the image domain after reconstruction of the sinogram projection data [104] [105]. This requires image reconstruction for every temporal frame in the conventional sampling schedule (typically 22 or more), which imposes a large computational burden and introduces reconstruction errors that can affect the later PCA. An alternative is to apply PCA early to the sinograms, before image reconstruction [89, 106, 107], for reducing the computational cost of image reconstruction and improving quantification. In [90], a combined temporal and spatial compression technique has been proposed for the compression of dynamic functional image data in sinogram domain, including a temporal compression stage based on the application of PCA directly to the sinogram data to reduce the dimensionality of the data, followed by a spatial compression stage using JPEG2000 to each principal component (PC) channel weighted by the signal in each channel. Figure 3.22 illustrates the framework of the combined data compression technique using PCA and channel-weighted JPEG2000 in sinogram domain, including the following four major coding steps: (1) Sinogram noise normalization; (2) Sinogram-domain PCA; (3) PCA channel weighting; (4) Sinogram-domain JPEG2000 coding, and three major decoding steps: (1) JPEG2000 decoding; (2) PCA channel reconstruction; and (3) Inverse PCA.

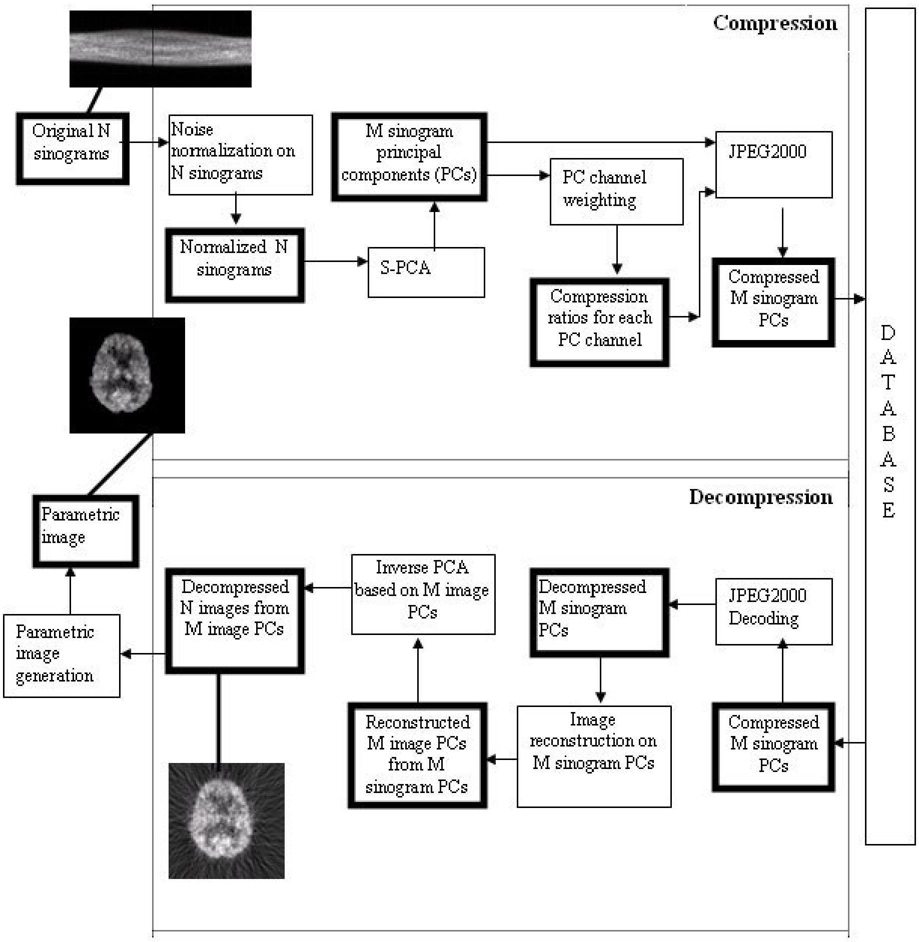

Compression
Original N sinograms
Noise normalization on N sinograms
M sinogram principal components (PCs)
JPEG2000
PC channel weighting
Normalized N sinograms
S-PCA
Compression ratios for each PC channel
Compressed M sinogram PCs
DATABASE
Decompression
Parametric image
Decompressed N images from M image PCs
Inverse PCA based on M image PCs
Decompressed M sinogram PCs
JPEG2000 Decoding
Parametric image generation
Reconstructed M image PCs from M sinogram PCs
Image reconstruction on M sinogram PCs
Compressed M sinogram PCs

Figure 3.22: Dynamic functional image data compression using PCA and channel-weighted JPEG2000 in sinogram domain [90]

Sinogram noise normalization: A PET scanner in 2D mode outputs the initial projection data in the form of *N* sinograms. This sinogram data is assumed to have been corrected for attenuation, randomness and differences in detector efficiencies. As PCA is a data driven technique that cannot itself distinguish noise from signal, it is necessary to scale or normalize each frame so that each has approximately equal noise variance [108]. Assuming approximate Poisson statistics (attenuation and random-corrected sinograms are no longer exactly Poisson), each temporal sinogram frame is noise-normalized through dividing by $\sqrt{N}/\Delta T$, where *N* is the total number of detection counts in the sinogram frame, and ΔT is the time duration of the frame. Several other data preprocessing transformations have been developed to normalize the noise in each frame to improve the PCA of the signal [109].

Sinogram-domain PCA: The PCA is applied directly to the time series of N (noise-normalized) sinograms to produce a reduced number of M sinogram principal component (S-PC) channels, and performed simultaneously on the data from all spatial planes. The objective of PCA is to represent orthogonal maximum variance directions for the analysed data set. This multivariate image analysis approach is well suited to high dimensional, highly correlated data such as dynamic PET data. If there are N frames in the conventional sampling schedule, PCA produces M principal components, where $M \leq N$ and the eigenvalues of the PCA channels are ordered from largest to smallest. Given a random vector population $\mathbf{X}_{sinogram} = (x_1, \ldots, x_\mathrm{n})^T$, $x_1, \ldots, x_\mathrm{n}$ in this case represents the individual time samples of the dynamic tomographic study in the sinogram domain, and the mean vector of the population is defined as $\mu_{sinogram} = E\{\mathbf{X}_{sinogram}\}$, and the covariance matrix is:

$$C = E\{(\mathbf{X}_{sinogram} - \mu_{sinogram})\,(\mathbf{X}_{sinogram} - \mu_{sinogram})^T\}. \tag{3.44}$$

After eigen-analysis of C, the eigenvalue-eigenvector pairs (λ_1, e_1), (λ_2, e_2), …, (λ_n, e_n) are ordered by eigenvalues in descending order. To reduce the data set, only the first M eigenvectors instead of using all the eigenvectors of the covariance matrix are used to represent the data; Let $\mathbf{A}_{sinogram}$ be a matrix consisting of the first M eigenvectors of the covariance matrix as the row vectors. The transformation of data vector $\mathbf{X}_{sinogram}$ is then given by:

$$\mathbf{P}_{sinogram} = \mathbf{A}_{sinogram}\,(\mathbf{X}_{sinogram} - \mu_{sinogram}) \tag{3.45}$$

where $\mathbf{P}_{sinogram}$ is in an orthogonal coordinate system defined by the eigenvectors.

PCA channel weighting: It is required that each principal component in the M sinogram principal components is compressed with different quality according to their priority (importance of the signal) in the set of principal components. The principal component channel with the higher priority requires less compression ratio.

Sinogram-domain JPEG2000 coding: JPEG2000 is applied to each M sinogram principal component channels with a weighted compression ratio to produce compressed M sinogram principal components. As mentioned in Section 3.2.8, JPEG2000 is based on the discrete wavelet transform (DWT) [36] rather than the discrete cosine transform (DCT), which provides significant improvements over JPEG including progressive decoding by image quality and improved compression efficiency.

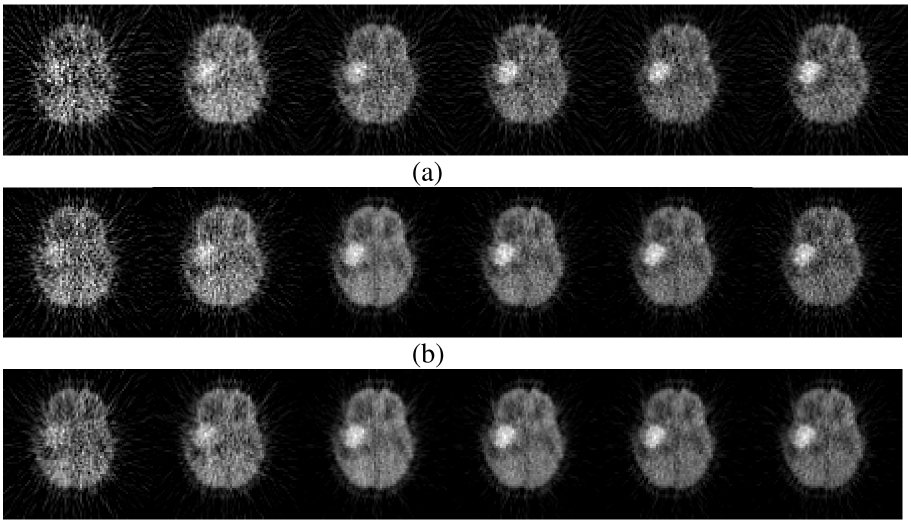
JPEG2000 decoding: The decoding is performed on the M JPEG2000 compressed sinogram principal component channels to regenerate M decompressed sinogram principal component channels.

PCA channel reconstruction: A set of M image-domain principal component channels is reconstructed from the M decompressed sinogram-domain principal components by using an image reconstruction algorithm such as filtered-back projection (FBP) or ordered sub-set expectation-maximization (EM). Define this reconstructed space as $\mathbf{R}_{image}$. An advantage of applying the PCA before the image reconstruction stage is that the lowered noise levels in the PCA channels allow for reduced filtering in the FBP algorithm, which reduces blurring and partial volume effects in the final result.

Inverse PCA: The inverse of the PCA is performed on the *M* image principal component channels to regenerate a time series of *N* image frames. In the step of "Sinogram-domain PCA", the PCA of the sinograms created a transformation matrix $\mathbf{A}_{sinogram}$ and a mean vector $\mu_{sinogram}$. The inverse PCA transformation in the image domain is then:

$$\mathrm{Inv}(\mathbf{R}_{image}) = (\mathbf{A}_{sinogram})^T \times \mathbf{R}_{image} + \mu_{sinogram} / N_{proj} \qquad (3.46)$$

where N_{proj} is the total number of projection angles in the tomograph. Finally, the inverse of the noise normalization weighting is performed.



(a)

(b)

(c)

Figure 3.23: The results of dynamic reconstructed PET series using different data compression methods: (a) the 11th, 13th, 15th, 17th, 19th and the last frame of original 22 temporal frames of CSS; (b) the corresponding reconstructed images from the PCA-only compression approach; (c) the corresponding reconstructed images from the combined compression approach based on PCA and channel-weighted JPEG2000 [90]

The results of performance evaluation demonstrate that the noise-normalized PCA can give equivalent compression ratios to OISS (to approximately five frames) but with approximately twice the precision. PCA in sinograms avoids introducing image reconstruction errors into the analysis, decreases the computational burden of image reconstruction and gives similar quantitative accuracy to OISS, and better accuracy than image-domain PCA. Figure 3.23 shows the results of dynamic reconstructed PET series using different data compression methods, in which the signal-to-noise ratio (SNR) of the reconstructed images from both PCA-only and the combined approaches is significantly improved. The improved SNR can be achieved because after applying noise normalization, the PCA can separate the signal from the noise. It is noted that the reconstructed images from the combined approach are slightly less noisy than that from PCA-only approach, because JPEG2000 itself has a denoising effect. The results indicate that the combined approach not only can reduce the quantity of data in dynamic PET, but can also improve the image quality of PET. Overall, the combined temporal and spatial compression technique for the compression of dynamic functional image data in sinogram domain can achieve a compression ratio as high as 129:1 while simultaneously reducing noise and improving physiological parameter estimation compared with the uncompressed data, and preserving the sinogram data for later analysis [90].

3.5 SUMMARY

The basics of picture compression are reviewed in this chapter. Information entropy dictates the minimum amount data needed to carry a certain amount of information. Any piece of data that exceeds the entropy level therefore contains redundancies. The purpose of compression is to remove these redundancies in data. The most rudimentary method for compressing data is through entropy coding. Entropy coders compress data to its source entropy using a variety of algorithms such as Huffman codes or arithmetic codes. Most competitive compression strategies employ predictive coding and context-based coding prior to entropy coding to further enhance compression performance.

Compression of pictures is performed either in lossless or lossy manner. Lossless compression is desirable since it maintains information integrity. However, it has limited compression ratio. Consequently, it has been more practical in many instances to apply lossy compression. For lossy compression, while it has superior compression performance over it lossless counterpart, it does not preserve all information contained within pictures. In addition, the lost in information is a function of the compression rate as governed by the rate-distortion function. Hence, for lossy coding, it is important to maintain a balance between compression and distortion/quality. The fidelity of pictures is dependent on how much perceptible information is contained within them. The purpose of perceptual coding is to compress pictures based on perceptible information. Consequently, the performance of perceptual coders is generally superior to non-perceptual coders in terms of visual quality. Perceptual coders operating at just above the super-threshold vision level produce picture quality that is imperceptible to the original pictures by only removing visually redundant information.

Compression of natural images is a straightforward task that juggles between compression ratio and quality. For medical images, lossy compression has been a sensitive issue primarily because it leads to the deterioration of information. However, a complete dependency on lossless compression is impractical due to its limited compression performance. To mitigate between information loss and compression performance, the idea of preserving diagnostic information as oppose to all information is gaining support. The challenge of this approach then is to devise a method for identifying diagnostic information. If diagnostics information is dependent on perception, then perceptual coding may be a solution to preserving diagnostic value of medical images while maintaining high compression performance. It is important to note that in medical compression, it is impossible to rely entirely on either lossless or lossy coding alone. Therefore, advanced data compression techniques used in noisy medical image data sets with high compression ratio and improved image qualities have been discussed in this chapter. These techniques have pioneered the biomedical diagnostically lossless data compression research. Three examples: compression in temporal domain, spatial domain, and sinogram domain, are used. These examples have demonstrated a great improvement of image quality and a significant reduction of image size for these new techniques, which have filtered out the measurement noise and provided much more reliable and smaller data sets ready for accurate diagnosis.

Acknowledgement

The authors are grateful to the support from ARC, PolyU/UGC grants. The authors thank Mr. David Wu for providing illustrations of compressed medical images.

3.6 EXERCISES

3.6.1 Given the follow probability distribution (below) for alphabet *A*

i	1	2	3	4	5	6	7
A(*i*)	*a*	*b*	*c*	*d*	*e*	*f*	*g*
p(*i*)	0.07	0.2	0.1	0.05	0.25	0.13	0.2

a) Calculate the entropy of *A*
b) Calculate fixed length code and compute data redundancy
c) Generate the Huffman tree and Huffman codes
d) Generate the arithmetic code number for sequence {a, e, g, c}

3.6.2 What are the advantages and disadvantages of lossless and lossy coding?

3.6.3 What is the purpose of predictive coding and context coding?

3.6.4 What are the basic components of lossy transform based coding? What is the purpose of each?

3.6.5 How does aliasing occur?

3.6.6 What is perceptual coding?

3.7 REFERENCES

1. ITU-R Recommendation BT.601-5, "*Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-Screen 16:9 Aspect Ratios*", ITU-T, 1995
2. N. S. Jayant and P. Noll, *Digital Coding of Waveform: Principles and Applications to Speech and Video*, NJ: Prenticle Hall, 1984
3. R. Clarke, *Transform Coding of Images*, Academic Press, 1985
4. A. K. Jain, “Fundamentals of Digital Image Processing”, NJ: Prentice Hall, 1989
5. R. Clarke, *Digital Compression of Still Images and Video*, Academic Press, 1995
6. A. N. Netravali and B. G. Haskell, *Digital Pictures – Representation, Compression and Standards*, New York: Plenum Press, 2nd eds., 1995
7. K. R. Rao and J. J. Hwang, “Techniques and Standard for Image, Video & Audio Coding”, NJ: Prentice Hall, 1996
8. K. Sayood, *Introduction to Data Compression*, CA:Morgan Kauffman Publishers, 3rd Ed., 2006
9. C. E. Shannon, “A Mathematical Theory of Communication”, *Bell System Technical Journal*, 27:379-423,623-656, July, October, 1948.
10. B. G. Haskell. A. Puri and A. N. Netravali, *Digital Video: An Introduction to MPEG-2*, NY: Chapman and Hall, 1997
11. D. A. Huffman, “Method for the Construction of Minimum Redundancy Codes”, P*roceedings of the IRE*, 40(9):1098-1101, Sept. 1952
12. J. Rissanen, “Generalised Kraft Inequality and Arithmetic Coding”, *IBM Journal of Research and Development*, 20:198-203, 1976
13. A. Moffat, R. Neal and I. H. Witten, “Arithmetic Coding Revisited”, *ACM Transaction on Information Systems*, 16(3):256-294, July,1998
14. X. Wu and N. Memon, “Context-Based Adaptive Lossless Image Coding*”, IEEE Transaction on Communications*, 45(4):437-444, April, 1997
15. ISO/IEC JTC 1/SC 29, “Information Technology -- JPEG 2000 image coding system -- Part 1: Core coding system”, ISO/IEC 15444-1:2000, 2000
16. ITU-T Recommendation H.264/ISO IEC 11496-10, “*Advance Video Coding”*, ITU-T/ISO, 2002
17. C. C. Cutler, “Differential Quantization of Communication Signals”, US Patent, No. 2,605,361, July, 1952
18. C. W. Harrison, “Experiment with Linear Prediction in Television”, *Bell Systems Technical Journal*, 29:764-783, 1952
19. W. M. Goodall, “Television by Pulse Coding Modulation”, *Bell Systems Technical Journal*, 28:33-49, Jan. 1951
20. M. Weinberger, G. Seroussi and G. Sapiro, “The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS”, *IEEE Transactions on Image Processing*, 9(8):1309-1324, August, 2000
21. C. E. Shannon, “Prediction and Entropy of Printed English”, *Bell System Technical Journal*, 30:50-64, Jan. 1951
22. ISO/IEC JTC 1/SC 29, “*Information technology -- Lossless and near-lossless compression of continuous-tone still images: Baseline*”, ISO/IEC 14495-1:1999, 1999
23. S. W. Golomb, “Run-length Encodings”, *IEEE Transaction on Information Theory*, IT-12:399-401, 1966
24. J. Capon, “A Probabilistic Model for Run-Length Coding of Pictures”, *IRE Transaction on Information Theory*, IT-5:157-163, Dec., 1959
25. R. Dansereau and W. Kinsner, “Perceptual Image Compression Through Fractal Surface Interpolation”, *Canadian Conf. on Electrical and Computer Engineering*, 2:899-902, 1996
26. J. E. Jacquin, “A Novel Fractal Block-Coding Technique for Digital Images”, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 4:2225-2228, 1990
27. M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*, Englewood Cliffs, NJ: Prentice-Hall, 1995
28. A. V. Oppenheim, R. W. Schafer and J. R. Buck, *Discrete-Time Signal Processing*, Upper Saddler River, NJ: Prentice Hall, 1999
29. S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*, NY: McGraw-Hill, 2nd Ed., 2001
30. J. W. Woods and S. D. O'Neil, “Subband Coding of Images”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(5):1278-1288, Oct. 1986
31. K. R. Rao and P. Yip, *Discrete Cosine Transform – Algorithms, Advantages and Applications*, Academic Press, 1990

32. P. J. Burt and E. H. Adelson, "The Laplacian Pyramid as a Compact Image Coder", *IEEE Transactions on Communication*, 31(4):532-540, April, 1983

33. S. G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674-693, July, 1989

34. J. Fourier, "*Théorie Analytique de la Chaleur*", 1822

35. V. R. Algazi and D. J. Sakrison, "On the Optimality of the Karhunen-Loève Expansion", *IEEE Transactions on Information Theory*, IT-15:319-321, Mar. 1969

36. D. Taubman and M. Marcellin, eds., "*JPEG2000: Image Compression Fundamentals, Standards and Practice*", Springer, ISBN: 0-7923-7519-X, 2002

37. S. G. Mallat and Z. Zhang, "Matching Pursuits with Time-Frequency Dictionaries", *IEEE Transaction on Signal Processing*, 41:3397-3415, Dec. 1993

38. S. Mitra and J. F. Kaiser, *Handbook of Digital Signal Processing*, John Wiley & Sons, 1993

39. S. P. Lloyd, "Least Square Quantization in PCM", *IEEE Transaction on Information Theory*, IT-28(2):127-135 , Mar. 1982

40. J. Max, "Quantizing for Minimum Distortion", *IRE Transaction on Information Theory*, 6(1):7-12, 1960

41. A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kulwar Academic Publishers, 1992

42. ISO/IEC JTC 1/SC 29, "*Information technology -- Generic coding of moving pictures and associated audio information: Video*", ISO/IEC 13818-2:2000, 2000

43. ISO/IEC JTC 1/SC 29, "Information Technology -- JPEG 2000 image coding system -- Part 3: Motion JPEG 2000", ISO/IEC 15444-3:2002, 2002

44. J. R. Ohm, "Three-dimensional Subband Coding with Motion Compensation", *IEEE Transaction on Image Processing*, 3(5): 559 - 571, Sept. 1994

45. D. Taubman and A. Zakhor, "Multirate 3-D Subband Coding of Video" *IEEE Transaction on Image Processing*, 3(5):572-588, Sept., 1994

46. J. Xu, Z. Xiong, S. Li, and Y.-Q. Zhang, "3-D Embedded Subband Coding with Optimal Truncation (3-D ESCOT)," *Applied and Computational Harmonic Analysis: Special Issue on Wavelet Applications in Engineering*, 10:290-315, May, 2001

47. J. Hua; Z. Xiong and X. Wu, "High-Performance 3-D Embedded Wavelet Video (EWV) Coding", *Proceedings of IEEE 4th Workshop on Multimedia Signal Processing*, 569 – 574, Oct., 2001

48. S.-T. Hsiang and J. W. Woods, "Embedded Video Coding Using Invertible Motion Compensated 3-D Subband/Wavelet Filter Bank", *Signal Processing: Image Communication*, 16(8):705-724, 2001

49. M. Benetiere, V. Bottreau, A. Collet-Billon and T. Deschamps, "Scalable Compression of 3D Medical Datasets Using a (2D+T) Wavelet Video Coding Scheme", *International Symposium on Signal Processing and its Applications*, 2:537-540, Aug. 2001

50. J. Xin, C. W. Lin, M. T. Sun, "Digital Video Transcoding", *Proceedings of the IEEE*, 93(1):84-97, Jan. 2005

51. D. Taubman, "High Performance Scalable Image Compression with EBCOT", *IEEE Transaction on Image Processing*, 9(7):1158-1170, Jul. 2000

52. J. R. Ohm, "Advances in Scalable Video Coding", *Proceedings of the IEEE*, 93(1):42-56, 2005

53. B. Girod, "What's Wrong with Mean-Squared Error". In A. B. Watson (Ed.), *Digital Images and Human Vision*, MIT Press, 1993

54. J. J. Hwang, H. R. Wu and K. R. Rao, "Picture Coding and Human Visual System Fundamentals", In H. R. Wu and K. R. Rao, eds., *Digital Video Image Quality and Perceptual Coding*, FL:CRC, pp. 3-43, 2006

55. B. A. Wandell, *Foundations of Vision*, Sunderland, MA: Sinuar, 1995

56. G. E. Legge and J. M. Foley. "Contrast Masking in Human Vision". *Journal of Optical Society of America, 70(12)*:1458-147l, Dec., 1980

57. A. B. Watson and J. A. Solomon, "Model of Visual Contrast Gain Control and Pattern Masking", *Journal of Optical Society of America A*, 14(9):2379-2391, Sept. 1997

58. E. P. Simoncelli, W. T. Freeman, E. H. Adelson and D. J. Heeger. "*Shiftable Multi-Scale Transforms". IEEE Trans. Information Theory*, 38(2):587-607, March 1992

59. A. B. Watson, "The Cortex Transform: Rapid Computation of Simulated Neural Images", *Computer Vision, Graphics, and Image Processing*, 39:311-327, 1987

60. D. M. Tan, H. R. Wu and Z. Yu, "Perceptual Coding of Digital Monochrome Images", *IEEE Signal Processing Letters*, 11(2):239-242, Feb. 2004

61. Z. Liu, L. J. Karam and A. B. Watson", “JPEG2000 Encoding with Perceptual Distortion Control", *IEEE Transactions on Image Processing*, 15(7):1763-1778, July, 2006
62. J. Limb, “On the Design of Quantisers for DPCM Coders - A Functional Relationship Between Visibility, Probability and Masking”, *IEEE Transactions on Communications*, 26:573-578, 1978
63. A. Netravali and B. Prasada, “Adaptive Quantisation of Picture Signals Using Spatial Masking”, *Proceedings of the IEEE*, 65(4):536-548, April, 1977
64. A. B. Watson, “DCT Quantization Matrices Visually Optimized for Individual Images”, *Proceedings of Human Vision, Visual Processing, Digital Display IV*, 202-216, 1993
65. T. P. O'Rourke and R. L. Stevenson, “Human Visual System Based Wavelet Decomposition for Image Compression”, *Journal of Visual Communication and Image Representation*, 6(2):109-121, June, 1995
66. D. Shkarin, “PPM: One Step to Practicability”, *Proceedings of the Data Compression Conference*, 202-211, 2002
67. I. Matsuda, H. Mori and S. Itoh, “Lossless Coding of Still Images using Minimum-Rate Predictors”, *Proceedings of IEEE International Conference on Image Processing*, 1:132-135, 2000
68. W. B. Pennekaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*, NY: Van Nostrand Reinhold, 1993
69. D. Santa-Cruz, R. Grosbois and T. Ebrahimi, “JPEG2000 Performance Evaluation and Assessment”, *Signal Processing: Image Communication*, 17(1):113-130, 2002
70. G. J. Sullivan, “Video Compression — From Concepts to the H.264/AVC Standard”, *Proceedings of the IEEE*, 19(1):18-31, Jan. 2005
71. D. Wu, D. M. Tan, M. Baird, J. DeCampo, C. White and H. R. Wu, “Perceptually Lossless Medical Image Coding”, *IEEE Transactions on Medical Imaging*, 25(3):335-344, March 2006
72. J. H. C. Reiber, G. Koning, J. Dijkstra, A. Wahle, B. Goedhart, F. H. Sheehan and M. Sonka, “Angiographic and Intravascular Ultrasound”, In the *Handbook of Medical Imaging*, J. Beutel, H. L. Kundel and R. L. van Metter, eds., 2nd Ed., SPIE Press, Bellingham, 2004
73. D. A. Clunie, “Lossless Compression of Grayscale Medical Images: Effectiveness of Traditional and State-of-the-art Approaches”, *Proceedings of SPIE Medical Imaging*, 3980:74-84, May, 2000
74. NEMA PS 3, “*Digital Imaging and Communications in Medicine* (*DICOM*) – Part 5: Data Structures and Encoding”, National Electrical Manufacturers Association, Virginia, 2006
75. A. J. Duerinckx and E. J. Pisa. “Filmless Picture Archiving and Communication System (PACS) in Diagnostic Radiology”, *Proceeding of SPIE*, 318:9-18, 1982. Reprinted in *IEEE Computer Society Proceedings of PACS'82*, order No 388
76. S. J. Dwyer III. “A personalized view of the history of PACS in the USA”. *Proceedings of the SPIE*: *Medical Imaging 2000: PACS Design and Evaluation: Engineering and Clinical Issues*, G. J. Blaine and E. L. Siegel eds., 3980:2-9, 2000
77. ISO/IEC JTC 1/SC 29, “*Information technology -- Digital compression and coding of continuous-tone still images: Requirements and guidelines*”, ISO/IEC 10918-1:1994, 1994
78. S. H. Becker and R. L. Arehson, “Costs and Benefits of Picture Archiving and Communication System”, *Journal of the American Medical Informatics Association*, 1(5):361-371, Sept./Oct. 1994
79. N. C. Phelan and J. T. Ennis, “Medical Image Compression Based on a Morphological Representation of Wavelet Coefficients”, *Medical Physics*, 26(8):1607-1611, Aug., 1999
80. O. Kocsis, L. Costaridou, L. Varaki, E. Likaki, C. Kalogeropoulou, S. Skiadopoulos and G. Panayiotakis, “Visually lossless threshold determination for microcalcification detection in wavelet compressed mammograms”, *European Radiology*, 13(10):2390-2396 , Oct., 2003
81. B.-J. Kim and W. A. Pearlman, “An Embedded Wavelet Video Coder Using Three-Dimensional Set Partitioning in Hierarchical Trees (SPIHT)”, *Proceedings of the Data Compression Conference*, 251-260, March, 1997
82. M. Krol, “Telemedicine”, *IEEE Potentials*, 16(4):29-31, Oct/Nov.1997
83. D. Feng, D. Ho, H. Iida and K. Chen, “Techniques for Functional Imaging”, an invited chapter contributing to *Medical Imaging Techniques and Applications*, Edited by: C. T. Leondes, in 'Gordon and Breach International Series in Engineering, Technology and Applied Science', *Gordon and Breach Science Publishers*, pp.85-145, 1997
84. D. Feng, “Information Technology Applications in Biomedical Functional Imaging”, *IEEE Trans. on Information Technology in Biomedicine*, vol. 3, no. 3, pp. 221-230, September 1999

85. X. Li, D. Feng and K. Chen, "Optimal Image Sampling Schedule: A New Effective Way to Reduce Dynamic Image Storage Space and Functional Image Processing Time", *IEEE Transactions on Medical Imaging,* vol. 15, no.5, pp. 710-719, October 1996
86. D. Ho, D. Feng and K. Chen, "Dynamic Image Data Compression in Spatial and Temporal Domains: Theory and Algorithm," *IEEE Trans. on Information Technology in Biomedicine*, vol. 1, no. 4, pp219-228, December 1997
87. D. Feng, W.Cai and R. Fulton, "Dynamic Image Data Compression in the Spatial and Temporal Domains: Clinical Issues and Assessment, " *IEEE Trans. on Information Technology in Biomedicine*, vol. 6, no. 4, pp262-268, 2002
88. Z. Chen, D. Feng, W. Cai and R. Fulton, "Performance Evaluation of Functional Medical Imaging Compression via Optimal Sampling Schedule Designs and Cluster Analysis", *IEEE Transactions on Biomedical Engineering*, vol.52, no.5, pp943-945, May 2005
89. Z. Chen, B. Parker, D. Feng and R. Fulton, "Temporal Processing of Dynamic Positron Emission Tomography via Principal Component Analysis in the Sinogram Domain", *IEEE Transactions on Nuclear Science*, vol.51, no.5, pp2612-2619, Oct 2004
90. Z. Chen, D. Feng and W. Cai, "Temporal and Spatial Compression of Dynamic Positron Emission Tomography in Sinogram Domain", *International Journal of Image and Graphics*, vol.5, no.4, pp839-858, 2005
91. Z. Chen, "Biomedical Functional Imaging Data Compression and Analysis", *PhD Thesis*, 2003
92. J. Delforge, A. Syrota and B.M. Mazoyer, "Experimental design optimization: Theory and application to estimation of receptor model parameters using dynamic positron emission tomography", *Physics in Medicine & Biology*, vol.34, pp419-435, 1989
93. R.A. Hawkins, M.E. Phelps and S.C. Huang, "Effects of temporal sampling, glucose metabolic rates, and disruptions of the blood-brain barrier on the FDG model with and without a vascular compartment: studies in human brain tumours with PET, *J. Cereb. Blood Flow Metab.* vol.6, pp170-183, 1986
94. S. Jovkar, A.C. Evans, M. Diksic, H. Nakai and Y.L. Yamamoto, "Minimization of parameter estimation errors in dynamic PET: Choice of scanning schedules, *Physics in Medicine & Biology*. vol.34, pp895-908, 1989
95. B.M. Mazoyer, R.H. Huesman, T.F. Budinger and B.L. Knittel, Dynamic PET Data Analysis, *J. Computer Assisted Tomography*. vol.10, pp645-653, 1986
96. X. Li and D. Feng, "Toward the reduction of dynamic image data in PET studies", *Computer Methods and Programs in Biomedicine*. vol.53, pp71-80, 1997
97. D.Z.D' Argenio, "Optimal sampling times for pharmacokinetic experiments," *Journal of Pharmacokinetics and Biopharmaceutics*, vol.9, pp.739-756, 1981
98. D. Feng, W. Cai and R. Fulton, "An Optimal Image Sampling Schedule Design for Cerebral Blood Volume and Partial Volume Correction in Neurologic FDG-PET Studies", *Australia New Zealand Journal of Medicine*, vol.28, no.3, pp361, 1998
99. D. Feng, X. Li, W.C. Siu, "Optimal Sampling Schedule Design for Positron Emission Tomography Data Acquisition", *Control Eng. Practice*. vol.5, no.12, pp1759-1766, 1997
100. L.D. Crocker, "PNG: the portable network graphic format," *Dr. Dobb's J.*, pp.36-49, July 1995
101. T. Thireou, L. G. Strauss, A. Dimitrakopoulou-Strauss, G. Kontaxakis, S. Pavlopoulos, A. Santos, "Performance evaluation of principal component analysis in dynamic FDG-PET studies of recurrent colorectal cancer", *Computerized Medical Imaging and Graphics,* vol. 27, pp. 43-51, 2003
102. A.H. Andersen, D.M. Gash, M.J. Avison, "Principal component analysis of the dynamic response measured by fMRI : a generalized linear systems framework." *Magnetic Resonance Imaging,* vol.17, no.6, pp. 795-815, 1999
103. B.S. Everitt and G. Dunn, *Applied Multivariate Data Analysis* (2nd ed.), Arnold, London, pp.48-65, 2001
104. V. Chameroy and R. D. Paola, "High compression of nuclear medicine dynamic studies", *International Journal of Cardiac Imaging* vol. 5, pp. 261-269, 1990
105. T. Kao, S.H. Shieh and L.C. Wu, "Dynamic Radionuclide Images Compression Based on Principal Components Analysis", *Engineering in Medicine and Biology Society*, vol. 3, pp. 1227-1228, 1992
106. C.M. Kao, J.T. Yap and M.N. Wernick, "High-resolution Reconstruction of Dynamic PET Image Sequences using a Low-order Approximation", *Proc. SPIE*, vol.2622, pp796-801, 1995
107. M. N. Wernick, E.J. Infusion and M. Milosevic, "Fast Spatio-temporal Image Reconstruction for Dynamic PET", *IEEE Transaction on Medical Imaging*, vol.18, pp185-195, Mar. 1999
108. F. Pedersen, M.B.E. Bengtsson and B. Langstrom, "Principal Component Analysis of Dynamic PET and Gamma Camera Images: A Methodology to Visualize the Signals in the Presence of Large Noise", *1993 IEEEE*

Conference Record Nuclear Science Symposium and Medical Imaging Conference, San Francisco, CA, vol.3, pp1734-38, 1993

109. M. Samal, M. Karny, H. Benali, W. Backfrieder, A. Todd-Pokropek and H. Bergmann, "Experimental comparison of data transformation procedures for analysis of principal components", *Phys. Med. Biol.*, vol. 44, pp. 2821-2834, 1999

110. I. H. Witten, R. M. Neal and J. G. Clear, "Arithmetic coding for data Compression". *Communications of the ACM,* vol.30, no.6, pp 520–540, June 1987

3.8 INDEX