SET REDUNDANCY, THE ENHANCED COMPRESSION MODEL, AND METHODS FOR COMPRESSING SETS OF SIMILAR IMAGES

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

by
Kosmas Karadimitriou

Department of Computer Science
Louisiana State University
Baton Rouge, LA 70803, USA
August 1996

ACKNOWLEDGMENTS

I am deeply indebted to my advisor Dr. John Tyler for his continuous support and encouragement throughout my graduate studies. His patience, enthusiasm, and generous contribution of time and resources have greatly helped me to achieve my goals. He has been more than my advisor, as he often made me feel part of his family. I am especially thankful for his trust in my ideas and abilities and for the opportunities he has offered me.

I would like to thank all the members of my graduate committee and particularly Dr. Iyengar, Dr. Kraft, and Dr. Carver for all the knowledge and support they offered me during my graduate studies at LSU.

I wish to thank and express my appreciation to my parents. They have given me direction in life, never letting me take the easy way, and they have offered me one of the most precious gifts: the values that have shaped my character.

I am particularly grateful to Maria for her immeasurable emotional and intellectual support throughout our graduate school years. She has played a major role in my decision to pursue a Ph.D. after my Master's degree. She has read all my manuscripts and provided thoughtful and constructive comments and criticisms, even though Computer Science is not her field. She has always been there to share my ideas, dreams, frustrations, and excitements.

Finally, I would like to thank Dr. Sidney Wallace, Dr. Marc Fenstermacher, and John Grossman from M.D. Anderson Cancer Center at Houston, Texas. They have given me the opportunity and the resources to test my theory and obtain the experimental results that are presented in this dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Image compression is the process of reducing the number of bits required to represent an image. This can be achieved by reducing (or ideally, eliminating) various types of redundancy that exist in the imaging data. All current image compression methods are based on the same theoretical model, which targets redundancy found in individual images. However, this model ignores an additional type of redundancy that exists in sets of similar images, the “set redundancy”. The source of set redundancy is the common information existing in more than one image in a set of similar images. Set redundancy can be recognized by the appearance of similar pixel values at the same image regions, comparable image histograms, similar image features, or analogous distributions of edges. This research explores the concept of set redundancy and establishes its importance for image compression. A new theoretical compression model is proposed, the Enhanced Compression Model, which extends the current model by including set redundancy extraction. The requirements and restrictions for set redundancy extraction are discussed, and practical methods to implement it are developed. These methods are collectively referred to as SRC or Set Redundancy Compression methods. Three SRC methods are presented: the Min-Max Differential (MMD) method, the Min-Max Predictive (MMP) method, and the Centroid method. According to the Enhanced Compression Model, the SRC methods can be combined with any current compression technique to achieve higher compression ratios when compressing sets of similar images.

One of the best application areas for the Enhanced Compression Model is medical imaging. Medical image databases usually store large sets of similar images and, therefore, contain significant amounts of set redundancy. Tests were performed by implementing the SRC methods on a test database of CT and MR brain images. The results show an average of as much as two-fold improvement in the performance of standard compression techniques when they are combined with the SRC methods. In addition, the SRC methods developed in this research are fast, lossless, and easy to implement.

CHAPTER 1

INTRODUCTION

As the beginning of the third millennium approaches, the status of the human civilization is best characterized by the term "Information Age". This term captures the essence of the transition from the physical world to the virtual, information-based societies. Information, the substance of this new world, despite its physical non-existence can dramatically change human lives. In our society, immaterial information is often worth more than gold. In a sense, being in the right place at the right time depends on having the right information, rather than just having luck.

Information is often stored and transmitted as digital data. However, the same information can be described by different datasets. The shorter the data description, usually the better, since people are interested in the information and not in the data per se. Data compression is the process of transforming the data description into a more succinct and condensed form. This improves storage efficiency, communication speed, and security.

The concept of data compression is surprisingly old. The first data compression applications can be traced back more than 2,000 years, to some of the first civilizations. For example, in ancient Greece the documents were written in a space-efficient form by removing all spaces between words. This conserved the costly papyrus and allowed for more economical storage of knowledge. In addition, Greeks used to appraise succinctness as a virtue; from early age, children were taught to speak "lakonika", meaning "concise" in the Greek language.

The value of data compression is evident when either the data-handling resources for storage or communication are scarce, or when the amount of data becomes overwhelmingly large. Unfortunately, people always tend to produce more data than they can efficiently handle, no matter how much data-handling capability they have. In the early days of personal computers, 10 MB of disk space were thought to be enough for most applications. Nowadays, 1,000 MB is often not enough; users request 2 GB or more in a standard PC configuration. In the communications arena the situation is similar. A few years ago, 300 baud rate was considered sufficient; today, a modem must be capable of at least 28,800 baud rate to compensate for data-hungry applications. This situation is the result of a moving target: the changing type of information-delivery media. Until quite recently, text was the basis of information delivery. When the technology matured for text-based applications, images were added, and the requirements increased. As soon as the technology could handle images efficiently, information delivery was augmented with video. In a few years, virtual reality applications will make the storage and communication requirements to soar again. This situation is responsible for the major importance of data compression technology.

Data compression is also important when electronics replace traditional media. For example, only a few years ago, medical imaging was completely film-based, with radiologists using exclusively films and viewboxes to make their diagnoses. Digital technologies were not part of routine examinations, and data compression was not important. However, the computer revolution has radically changed the medical environment, which is currently moving toward the goal of "filmless hospital". Digital systems are an integral part of CT, MRI, PET, SPECT, and Ultrasound imaging and even traditionally non-digital techniques (e.g., film X-rays) are gradually evolving to digitized imaging. All these digital imaging technologies create problems for storing, communicating and manipulating large amounts of data. An MRI or CT exam requires an average of 5 to 12 Mbytes of storage, and a single digitized X-ray film requires as much as 24 Mbytes. When thousands of exams must be stored digitally, the problem becomes critical. In this environment, data compression has become an indispensable tool.

Data compression techniques are based on the removal of redundant data. In monochrome images, three types of data redundancy can be found: interpixel, psychovisual, and symbol redundancy. These redundancies exist between the pixel values of an individual image. Current compression methods efficiently reduce these types of redundancy. However, in sets of similar images, an additional type of redundancy exists, which is referred here as "set redundancy".

This study introduces and investigates the concept of set redundancy (chapter 3), and proposes a new theoretical compression model that includes set redundancy reduction: the Enhanced Compression Model (chapter 4). Based on this model, three new compression methods are presented (chapters 5 and 6), which collectively are referred to as Set Redundancy Compression, or SRC methods. These methods can be used to improve compression in sets of similar images. "Similar images" are defined as images with comparable histograms and analogous spatial distributions of features. Based on this definition, many types of image databases store large sets of similar images. Medical image databases are an example in which the Enhanced Compression Model can be used very effectively. In this dissertation, application of the SRC methods on a small medical image database is presented (chapter 7). The experimental results show as much as a two-fold improvement in compression over existing image compression methods. These results confirm the existence of "set redundancy" in similar images, validate the Enhanced Compression Model as an improved theoretical model, and demonstrate the practical value of the proposed SRC compression methods.

CHAPTER 2

IMAGE COMPRESSION

2.1 INTRODUCTION

Data compression is defined as the process of encoding the data using a representation that reduces the overall size of the data. This reduction is possible when the original dataset contains some type of redundancy. *Digital image compression* is a field that studies methods for reducing the total number of bits required to represent an image. This can be achieved by eliminating various types of redundancy that exist in the pixel values.

The rapid development of imaging technologies has generated great interest in efficient compression methods. One reason for this interest is that digital images are stored as datasets which are often very large. Some examples of the storage volume required for different types of digital images are:

- low resolution color video frame: 0.8 Mbytes (512 x 512 pixels, 3 bytes per pixel)
- electronic or scanned photograph: 18 Mbytes (3000 x 2000 pixels, 3 bytes per pixel)
- digital medical X-ray image: 24 Mbytes (4000 x 4000 pixels, 12 bits per pixel)
- LANDSAT image: 216 Mbytes (6000 x 6000 pixels, 6 bytes per pixel)

The large amounts of data required to represent visual information often overtaxes the capabilities of current computer systems. Image compression can significantly help by reducing these data storage requirements. This in turn enables faster transmission of images and it can also provide enhanced data security. Moreover, it makes possible the development of efficient image processing algorithms that need less processing time by working directly with the compressed data (for example, Yeo and Liu [Yeo95] developed a method for volume rendering of DCT-compressed 3-dimensional data).

For these reasons, image compression is an important tool in the imaging environment. Furthermore, the role of image compression becomes increasingly important as the amount of imaging data produced per year grows exponentially, the number of imaging applications increases, and the transmission of digital images is already the largest portion of data traffic in many computer networks.

There are two basic types of compression: *lossless* and *lossy*. In lossless compression no information is lost, and decompressing the data always yields exactly the original data. In lossy compression however, the procedure is not reversible and the decompressed data only approximately match the original data. Generally, in image compression the lossy methods have better compression ratios than the lossless ones. However, lossy methods can result in degraded image quality. For some applications this tradeoff is appropriate (e.g., digital videoconferencing), but for others it can be unacceptable (e.g., applications in Medical Imaging).

In digital images three basic data redundancies can be identified and reduced: the *coding* redundancy, the *interpixel* (or *spatial*) redundancy, and the *psychovisual* redundancy. Color images also contain *spectral redundancy*, and video signals exhibit *temporal redundancy*.

Because of its importance, intensive research in the last half century has produced a large number of data compression methods. A review of the more well-known algorithms will be presented in this chapter, along with a brief description for each.

Before describing any compression methods, some terminology must be introduced. Every dataset is a sequence of *symbols*. This sequence can be considered to be the output of an *information source S*, capable of generating symbols taken from the *source alphabet* $\{a_0, a_1, ..., a_n\}$. The elements of the source alphabet are called *letters* or simply symbols. For every information source S, there is a probability $\Pr(a_j)$ that the source will produce the symbol a_j. *Entropy* is a measure of information, and it represents the minimum size of a dataset necessary to convey a particular amount of information (see Appendix A). Note that the terms “information” and “data” are not synonymous. *Data* are used to convey *information*, but there is one-to-many correspondence between a certain piece of information and the datasets that can be used to describe it. For example, to describe the number “one”, we can use “1”, or “01”, or “001”, etc. Clearly, there is no upper bound on the amount of data that can be used to describe the same piece of information. The difference between the size of a dataset and its entropy (theoretically its minimum size) gives a measure of the data redundancy present in the dataset.

Research in data compression has grown rapidly the last 50 years. This resulted in a large number of both lossless and lossy compression methods. It is difficult to create an exhaustive list of all techniques and their variations that have been developed throughout these years. Review papers (e.g., [Netravali80, Jain81a, Bassiouni85, Wong95]) usually provide an overview of the latest developments in this area and the major techniques used. In the same spirit, only some of the major image compression methods will be reviewed here. These methods are representative of research directions in data compression.

2.2 LOSSLESS COMPRESSION METHODS

2.2.1 Huffman Coding

Huffman coding is one of the oldest compression methods. It is based on data statistics, and it represents the symbols of the alphabet by variable code length, depending on their probability of occurrence (the more probable a symbol is, the shorter the code it is assigned). It was developed by Huffman [Huffman52] and is very similar to a technique developed earlier (1949) by Shannon and Fano. The difference between the two techniques is in the way the code for each symbol is constructed. Huffman coding constructs the code-generating tree bottom-up, starting from the leaves which correspond to symbol probabilities. The Shannon-Fano technique constructs the tree top-down, by recursively dividing the symbols into two groups of equal total probabilities. Huffman coding performs better than Shannon-Fano's technique, and actually Huffman proved that his technique was optimal under certain assumptions (which however do not always hold). There are many variations of Huffman's technique; two examples are the Modified Huffman Encoding [Hankamer79], and the Dynamic Huffman Encoding [Knuth85].

2.2.2 Arithmetic Coding

This method approaches very close the theoretical limit of compression efficiency. It is based on data statistics like Huffman's method, but it is usually more efficient. Its basic advantage is that unlike Huffman coding, it does not have the limitation that each codeword has to be at least one-bit long. However, its implementation is complicated and is very computationally intensive. The basic idea in arithmetic coding is to divide the interval between 0 and 1 into a number of smaller intervals corresponding to the probabilities of the message's symbols. Then the first input symbol selects an interval, which is further divided into smaller intervals. The next input symbol selects one of these intervals, and the procedure is repeated. In this way, the selected interval narrows with every symbol, and at the end, any number inside the final interval can be used to represent the message.

Arithmetic coding has evolved through the work of many researchers. The first names associated with it are Elias and Abramson (circa 1960). A theoretical description of this method was presented by Rissanen and Langdon [Rissanen79], whereas implementation issues can be found in [Langdon84]. A revised version of arithmetic coding has been developed recently, and it is based on the work of Moffat, Neal and Witten [Moffat95], and Witten, Neal and Cleary [Witten87]. A variation of arithmetic coding is the Q-coder, developed in IBM in the late 1980's [Pennebaker88, Mitchell88].

2.2.3 Lempel-Ziv Compression

Lempel-Ziv compression actually refers to two different approaches to dictionary-based compression: the LZ77 [Ziv77] and the LZ78 [Ziv78], both developed by Ziv and Lempel. In LZ77, the general idea is to have a “sliding window” which moves over the data. When a data sequence is encountered which already exists in the current window, then it is substituted by the pair {position, length} which points back to the existing sequence. LZ78 does not use this “sliding window”. Instead, it constructs dynamically a dictionary from the input file and then it replaces data sequences by their index in the dictionary.

Several compression methods have been developed based on these ideas, differing only on how they manage the dictionary. One of the most well known methods, the LZW (Lempel-Ziv-Welch), was designed by Welch in 1984 [Welch84]. This method is based on the LZ78 scheme, and it starts with a small dictionary (single letters only) expanding it with new strings during compression. The dictionary does not need to be stored, since it can be rebuilt during decompression.

2.2.4 Run Length Encoding

In its simpler implementation, Run-Length Encoding replaces repeated data by a {length, value} pair, where “value” is the repeated data value and “length” is the number of repetitions. This technique is especially successful in compressing bi-level images, and it is widely used in facsimile (FAX) systems. If direct application of Run-Length Encoding on a gray-scale image does not produce satisfactory compression, then the image can be decomposed into bit planes and every bit-plane can be compressed separately. There are many variations of Run-Length Encoding, for example [Tanaka82].

2.2.5 Differential Pulse Code Modulation (DPCM)

DPCM is an image compression method that is based on predictive coding. The basic idea is to predict the value of each pixel by using the values of its neighboring pixels, and then store only the prediction error. Typically, the errors are small, therefore fewer bits are required to store them. Depending on how many neighboring pixels are used, DPCM is classified as 1st order (1 pixel), 2nd order (2 pixels), 3rd order (3 pixels), etc. A prediction model that is often used for 3rd order DPCM is the following:

$$P(x,y) = r_1 \cdot P(x-1,y) - r_1 \cdot r_2 \cdot P(x-1,y-1) + r_2 \cdot P(x,y-1).$$

The values used for the coefficients r_1 and r_2 can vary, and usually they are image-dependent. A set of values that works well for many cases, and can simplify the calculations is $r_1 = r_2 = 1$. Usually, different regions in the same image have different optimal prediction coefficients. To account for this, adaptive prediction can be used, which splits the image into blocks and then computes independently the prediction coefficients for each block. Netravalli in [Netravali80] presented a review of predictive coding techniques.

2.2.6 Hierarchical Interpolation (HINT)

Hierarchical Interpolation [Roos88] is a multi-resolution coding scheme based on subsampling. It starts with a low resolution version of the input image, and it interpolates the pixel values to generate successively higher resolutions. The errors between the interpolation values and the real values are stored, along with the initial low-resolution image. Both the low-resolution image and the error values can be stored using fewer bits than the original image, therefore compression is achieved. The decompressor can reconstruct the original image using the same procedure, and adding the error values to the interpolation results.

2.2.7 Laplacian Pyramid

Laplacian Pyramid is a multi-resolution image compression method developed by Burt and Adelson [Burt83]. The basic idea is to construct successively lower resolution versions of the original image, and then find the differences between successive resolution versions. The lowest resolution image together with the difference images are sufficient to perfectly reconstruct the original image. Laplacian Pyramid has the drawback of increasing the number of data values by 4/3, therefore it is difficult to achieve high compression ratios, although the difference values can be stored very economically. On the other hand, this method can be used efficiently for progressive transmission of images.

2.2.8 Other Methods

Some other lossless compression methods include *Multiplicative Autoregression* (*MAR*) [Das93], *Bit-plane Encoding*, *Contour Encoding*, etc. Several methods have been developed for specific applications. For example, bi-level images (black and white) can be compressed by using *Constant Area Coding*, *White Block Skipping* (see [Gonzalez93] for more details), or *Irreducible Covers of Maximal Rectangles* [Cheng88]. Another example is compression in business or scientific databases, which includes methods such as *Multigroup Compression*, *Front/rear Compaction*, and *Null Bit-maps* [Bassiouni85]. Finally, there are techniques that although rank low in compression efficiency, they still have theoretical interest. An example is the binary representation of integers using Fibonacci numbers instead of powers of 2. The properties of this representation can be used for variable length encoding. However, this method compares poorly with other methods [Lelewer87].

2.3 LOSSY COMPRESSION METHODS

2.3.1 Quantization

Quantization is a many-to-one mapping that replaces a set of values with only one representative value. By definition this scheme is lossy, because after this mapping the original value cannot be recovered exactly. There are two basic types of quantization, the *scalar* and the *vector* quantization. Scalar quantization (SQ) performs many-to-one mapping on each value, for example, it may store only the 6 most significant bits from 8 bit values. Vector quantization (VQ) replaces arrays of values (i.e., blocks of pixels) with one value, which is the index from a “codebook”. The same index can be used to represent slightly different arrays of values, therefore it results in a lossy many-to-one mapping. VQ on images can cause edge distortion since edges cannot be reproduced perfectly by a small size codebook. VQ asymptotically outperforms SQ, but it is also more difficult to implement. The main implementation issues of VQ relate to codebook design, and to codebook search strategy. A widely used algorithm for codebook design is the LBG algorithm, developed by Linde, Buzo, and Gray [Linde80]. There is no standard algorithm for codebook searching, so different schemes are currently in use (for example, Full Search VQ, Tree-Structured VQ, Pruned Tree-Structured VQ, Entropy-Pruned Tree-Structured VQ, Entropy-Constrained VQ, etc.). A good review on VQ techniques is [Nasrabadi88].

2.3.2 Transform Coding

Transform coding is a general scheme for lossy image compression. It uses a reversible, linear transform (such as the Fourier transform) to map the image into a set of coefficients which are then quantized and coded. A good transform packs as much information as possible into a small number of transform coefficients. Then quantization selectively eliminates the coefficients that carry the least information. To simplify the computations, the image is usually divided into a number of small blocks, and the transform is applied separately to each of these blocks. There are numerous transforms that can be used, including Discrete Fourier (DFT), Discrete Cosine (DCT), Karhunen-Loeve (KLT), Walsh-Hadamard (WHT), etc. The one that is theoretically best, but usually not practical, is the KLT, which is based on the Hotelling Transform. The best alternative is the DCT which is used by most practical transform systems. Transform coding methods are suitable for progressive transmission of images. A rough approximation of the original image can be constructed with only a few coefficients, and it can be progressively refined as more coefficients are received. For a review of transform coding techniques, see [Wintz72].

2.3.3 Discrete Cosine Transform (DCT)

DCT is one of the most efficient transform coding schemes. It is also the base of the JPEG standard for image compression [Wallace91]. For an image block of size *L*x*L*, the 2-D DCT can be expressed as follows:

$$Y_{k,l} = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} x_{i,j} \cdot C_{k,l} \cdot \cos\frac{(2i+1)k\pi}{2L} \cdot \cos\frac{(2j+1)l\pi}{2L},$$

where:

(i,j) are coordinates of the pixels in the initial block,

(k,l) are coordinates of the coefficients in the transformed block,

$x_{i,j}$ is the value of the pixel in the initial block,

$Y_{k,l}$ is the value of the coefficient in the transformed block,

k = 0, 1, ..., L-1,

l = 0, 1, ..., L-1, and

$$C_{k,l} = \begin{cases} 1/2 & \text{, if } \; k = l = 0 \\ 1/\sqrt{2} & \text{, if } \; k = 0 \;\text{ or }\; l = 0 \\ 1 & \text{, if } \; k \neq 0 \;\text{ and }\; l \neq 0. \end{cases}$$

Some of the advantages of DCT over other transform coding methods are the use of real arithmetic, less computational complexity, the existence of fast DCT algorithms, and the availability of DCT chips. A disadvantage of DCT is the blocking (or tiling) artifacts that appear in high compression ratios. The same artifacts may appear in other transforms as well, because typically the transforms are implemented on pixel blocks, rather than the whole image. To avoid these artifacts, full-frame DCT can be used. However, the tradeoff is the increased computational requirements and the appearance of ringing artifacts (periodic patterns due to the quantization of high frequencies).

2.3.4 Subband Coding

In subband coding the input image is filtered through a set of operations that divide the input into a number of bands. The result is a number of sub-images with specific properties; for example, a smoothed version of the original, plus a set of images with the horizontal, vertical, and diagonal edges that are missing from the smoothed version. These sub-images can be compressed more efficiently than the original image, because the restricted type of information in each sub-image allows well-tailored encoding. Advantages over other compression methods include the lack of blocking artifacts, and the flexibility this scheme offers for adaptive compression. The most successful subband coding method is the wavelet decomposition, which is reviewed next.

2.3.5 Wavelet Decomposition

The theory of wavelets was developed in the mid-eighties and since then it has evolved into a valuable mathematical tool not only for image compression, but also for many other applications in signal processing. A simple definition of wavelets was given by Strang [Strang93]:

> "*A function W(x) is a wavelet if the translations and dilations of W(x) are mutually orthogonal*."

The function W(x) is called *mother wavelet*. The set of translations and dilations of W(x) forms an orthornormal basis, with good localization properties in both spatial and frequency domains. This basis can be used to approximate any function, in the same way Fourier series can approximate continuous signals. The advantage of wavelets over Fourier analysis is that wavelets have short duration, unlike the infinite duration of sines and cosines used in Fourier series. Consequently, wavelets are more efficient in representing local functions appearing in images.

There are many different wavelet functions. Perhaps the simplest one is the Haar wavelet, which is simply a square-shaped function. Because of its simplicity, the Haar wavelet has a poor approximation to most signals. More complicated wavelet functions provide better performance. For example, the Daubechies wavelets are highly irregular and resemble fractals. This irregularity allows them (by shifting and scaling) to represent even complex signals [Strang94].

The theory of wavelets was first developed by Morlet, Grossmann, and Meyer; Daubechies and Mallat are two other researchers with major contributions. The research is ongoing, and the number of papers published on wavelets grows at a fast rate. Some survey or tutorial papers on this subject are [Mallat89], [Rioul91], [Strang93], [Strang94].

2.3.6 Fractal Compression

Fractal compression is based on the theory of fractal geometry, developed in the seventies by Mandelbrot [Mandelbrot77]. Fractal objects have very complicated structures (in fact, infinitely complicated) which result from the recursive application of simple geometrical transformations. This property of fractals when applied to 2- dimensions can create realistic synthetic images using only a few rules and parameters. Barnsley [Barnsley89] was the first to propose the use of fractals in the inverse direction; that is, given a natural image, to find a set of rules and parameters that can create an approximation to this image. This can potentially lead to significant compression, because fractal-generating rules (called *Iterated Function Systems* or *IFS*) typically require only a fraction of the storage space of the original image. However, finding these rules is an inverse problem, which requires very large computational effort. The first practical fractal image compression method was presented by Jacquin [Jacquin92], and currently most of the fractal compression methods are based on his technique.

Fractal compression is one of the newest image compression methods with promising initial results. The theory of fractals has also inspired the development of methods that do not directly use *IFS*, but use the self-similarity of images under different scales to improve compression (for example, [Pentland93]).

2.3.7 Other Methods

There are many other methods for lossy image compression, that employ a wide variety of theories and techniques. Some of these methods are very interesting in their approach to the problem of compression. *Quadtree-based compression* recursively divides non-uniform image regions into four sub-regions, until satisfactory uniformity is reached; its goal is to encode economically large, uniform image regions. *Lapped Orthogonal Transform* performs a transform on overlapping image blocks, in order to avoid blocking artifacts. *Model-based coding* uses a priori models and stores only the set of parameters that transform these models into the image to be coded. *Block Truncation Coding* divides an image into blocks of pixels, and then encodes every block using only two values and a bitmap (1's and 0's) that indicates which value is used for each pixel. Finally, *Delta Modulation* is a simple technique that uses 1 bit/pixel and a step function that follows the direction of the slope in the image signal.

2.4 DISTORTION MEASURES

Lossy compression methods result in some loss of quality in the compressed images. It is often necessary to use some distortion measure to quantify this quality loss. Distortion measures commonly used are the *Root Mean Square Error (RMSE),* the *Normalized Mean Square Error (NMSE),* and the *Peak Signal-to-Noise Ratio (PSNR)*. These measures are defined as follows:

$$RMSE = \sqrt{\frac{1}{N \cdot M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left[f(i,j) - f'(i,j) \right]^2} ,$$

$$NMSE = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left[f(i,j) - f'(i,j) \right]^2 \Bigg/ \left[\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} f(i,j) \right]^2 , \text{ and}$$

$$PSNR = 20 \cdot \log_{10}\left(\frac{255}{RMSE}\right),$$

where the images have *N*x*M* pixels (8 bits per pixel), *f(i,j)* represents the original image, and *f '(i,j)* the reconstructed image after compression-decompression. Ad-hoc measures have also been proposed to measure specific distortions; for example, Saipetch *et.al.* [Saipetch95] used the *Normalized Nearest Neighbor Difference* to measure the intensity of blocking artifacts in DCT compression.

2.5 CONCLUSION

The list of data compression methods grows as new algorithms or variations of the already existing ones are introduced. All these data compression methods are based on the same principles and on the same theoretical compression model, which reduces effectively the three types of redundancy present in gray-level images: the interpixel, psychovisual, and symbol redundancy. However, sets of similar images contain an additional type of redundancy, that is not reduced by the current compression methods. In the next chapter this new type of redundancy is presented, and its connection with image compression is established.

CHAPTER 3

SET REDUNDANCY

3.1 THE CONCEPT OF "SET REDUNDANCY"

As stated previously, interpixel, psychovisual, and coding redundancy are the three types of redundancy in still monochrome images. However, in a *set of similar images*, one can observe a significant additional amount of *inter-image* redundancy. "**Similar images**" are images that have:

(a) similar pixel intensities in the same areas,
(b) comparable histograms,
(c) similar edge distributions,
(d) analogous distributions of features.

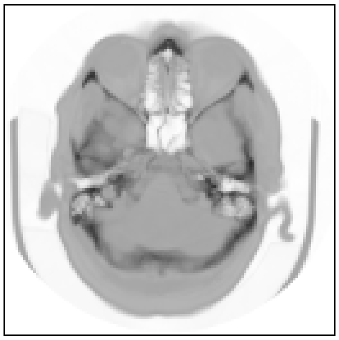
For example, consider a set of medical images produced by the same modality (e.g., CT) and depicting the same part of the human body (e.g., brain). These images have similar histograms, edge distributions, and features; in addition, their pixel intensities over the same areas are expected to be statistically similar. Therefore, according to the above definition, these images are statistically correlated and "similar" to each other.

The existence of statistical correlation between two images can be verified graphically with a scatterplot of pixel values, and numerically by calculating the correlation coefficient. The scatterplot can be obtained by plotting the points (x_i, y_i) where x_i is the value at pixel position P_i from image X, and y_i is the value at the same pixel position from image Y. If the images are correlated, then these points will cluster around the 45^o degree line passing through the origin. In fact, for perfectly correlated images (i.e., for identical images), these points form a perfect line with a 45^o slope. On the other hand, for two non-correlated images, there is no pattern around the 45^o line. The correlation coefficient can be used to quantify the statistical correlation. For two datasets $X = (x_1, x_2, \dots x_N)$ and $Y = (y_1, y_2, \dots y_N)$ with mean values x_m and y_m, the *linear correlation coefficient* is defined as [Neter89]:

$$r = \frac{\sum_{i=1}^{N}(x_i - x_m)(y_i - y_m)}{\sqrt{\sum_{i=1}^{N}(x_i - x_m)^2}\sqrt{\sum_{i=1}^{N}(y_i - y_m)^2}} \ .$$

This measure of correlation is also called the *product-moment correlation coefficient*, or *Pearson's r*. In order to avoid negative values, r^2 is often used instead of r. A value of r^2 close to 0 indicates that no correlation exists between the two datasets X and Y. A value approaching 1 shows the existence of a strong correlation. If $r^2 = 1.0$ then the datasets are perfectly correlated; in the case of images this means that the images are identical.

To demonstrate the existence of correlation among similar images, two pairs of images were considered. The first pair was the following two non-similar images:

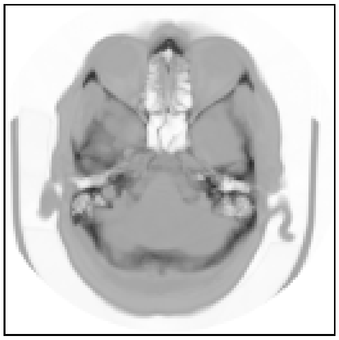


CT08

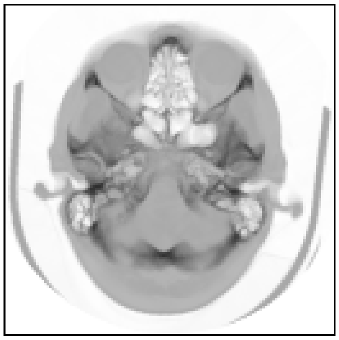
Lenna

The scatterplot resulting from comparing the values of these two images is shown on Figure 1. As we see, there is no clustering of values along the 45° line. Also the value of r^2 is 0.009, which is very close to zero (no correlation).

The second pair of images were the following two CT brain scans:



CT08



CT15

These two images are very similar, even though they are from different, non-related patients. Their scatterplot is shown on Figure 2. As we see, there is a noticeable clustering of values along the 45° line. Also the value of r^2 is 0.784, which shows the existence of a strong correlation.

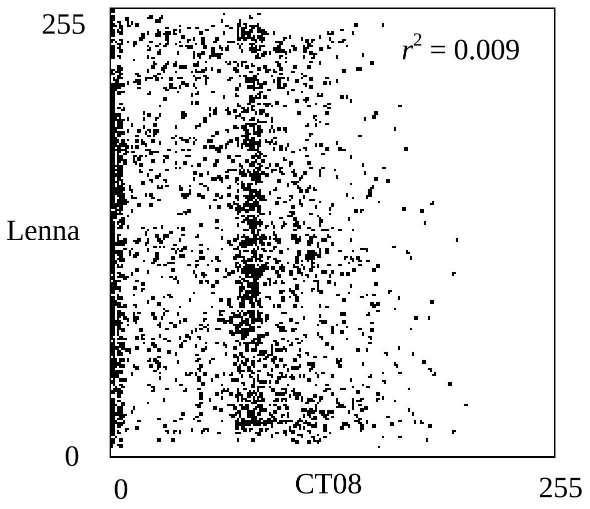


Figure 1: Scatterplot between two random images

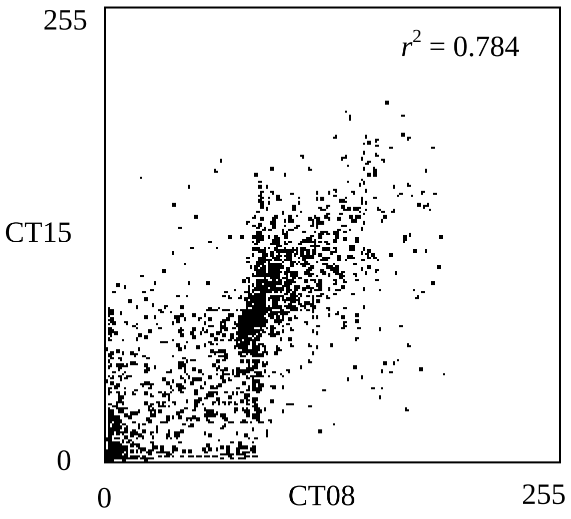


Figure 2: Scatterplot between two similar images

This statistical correlation among similar images is a result of inter-image redundancy. In this study, the term “set redundancy” is introduced to describe this type of redundant information, and is defined as follows:

Definition: ***Set redundancy** is the inter-image redundancy that exists in a set of similar images, and refers to the common information found in more than one image in the set.*

Set redundancy can be used to improve compression. A limit to compression is imposed by the image entropy. In the next section it is shown how set redundancy can be used to decrease the average image entropy in a set of similar images.

3.2 Image Entropy and Set Redundancy

The concept of *entropy* is defined in information theory as a measure of information [Shannon48]. The *image entropy* measures the amount of information an image contains (see Appendix A) and it is used as a measure of the compressibility of the image (lower entropy means better compressibility). The entropy of an individual image is calculated by using the histogram distribution of its pixel values, which represent “individual image statistics”. However, in a set of images, “set statistics” can be used instead of individual image statistics, resulting in smaller average image entropy. The reason is that in sets of similar images every pixel position (x, y) is associated with its own histogram distribution of gray values. By the definition of similarity among images (similar values at similar positions), these distributions will be highly non-uniform. If these distributions are used to encode the pixel values, then the entropy will be very small.

The decrease of image entropy resulting from the use of set statistics can be demonstrated mathematically. For example, consider a set of similar images with similarity S among the images. If $S = 0.4$ then, on the average, for every pixel position 40% of the pixel values across all images will have the same value. Note that some areas of the image may have higher variability than others; however, *on the average* the similarity will be 40%. Also, on the average, in every pixel position there is variability V among the images, where:

$$V = 1.00 - S.$$

Suppose that there are $(n+1)$ symbols and the alphabet is $\{a_0, a_1, a_2, \ldots , a_n\}$. For a specific pixel position let a_k be the symbol with the highest frequency and $\Pr(a_k)$ its probability of occurrence. Then:

$$\Pr(a_k) = S.$$

For simplicity, assume that the other symbols appear with equal probability, V / n :

$$\Pr(a_0) = \Pr(a_1) = ... = \Pr(a_{k-1}) = \Pr(a_{k+1}) = ... = \Pr(a_n) = V / n ,$$

so that,

$$\Pr(a_0) + \Pr(a_1) + ... + \Pr(a_n) = S + n\,(V / n) = S + V = 1.0 .$$

The entropy H is defined as [Shannon48]:

$$H = -\sum_{j=0}^{j=n} \Pr(a_j) \cdot \log(\Pr(a_j))$$

In our case, this entropy becomes:

$$H = -S \log(S) - \underbrace{(V/n)\log(V/n) - (V/n)\log(V/n) - ... - (V/n)\log(V/n)}_{n \text{ times}} ,$$

or,

$$H = - S \log(S) - V \log(V/n) , \qquad \text{(Eq. 1)}$$

where $n = 255$ for 8-bit gray-scale images. Figure 3 presents the values of entropy H resulting from (Eq. 1) for different values of S. The entropy clearly decreases as the similarity increases among the images in the set.

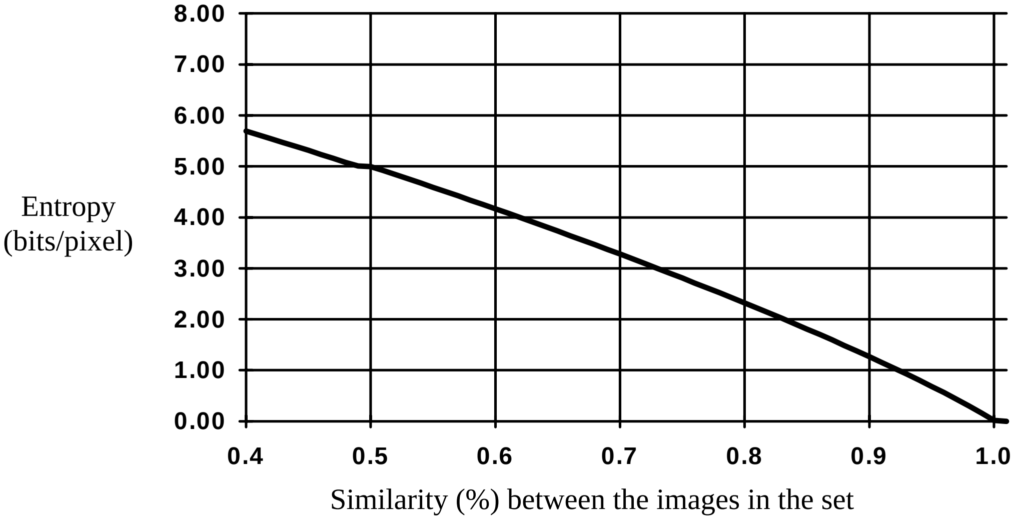


Figure 3: Entropy versus similarity in sets of similar images

Based on “individual image statistics”, the *first-order* entropy for 8-bit gray-scale photographic images is usually 5-7 bits/pixel. However, Figure 3 shows that the first-order entropy estimate can be smaller than this when “set statistics” are used. When the similarity is more than 40%, the average entropy of the images in the set is less than 6 bits/pixel. For 80% similarity, the entropy is 2.32 bits/pixel, and for 90% similarity is 1.27 bits/pixel. If higher-order entropy estimates are used, then the entropy will be even smaller. This clearly suggests that for sets of images with high similarity, it is advantageous to use set statistics instead of individual image statistics in compressing the images. Implicitly, the use of set statistics reduces the set redundancy from the images, resulting in better compression ratios. The compression can be further improved if set statistics are coupled with individual image statistics. In this case, each image in the set can be encoded using the “locally optimal” distribution; in other words, the statistics that produce the best compression.

The problem with the above schemes is that they require storing a histogram table for every pixel position. For images with 8 bits per pixel, this results in a 256-entry table per pixel. Unfortunately, it is not practical to keep so much statistical data for every pixel position. A practical method must reduce set redundancy while using only limited set statistics. The next chapter presents procedures that can reduce set redundancy using only limited set statistics, i.e., average, minimum, or maximum values. Also it establishes the theoretical framework for including set redundancy reduction into the general compression model.

CHAPTER 4

THE ENHANCED COMPRESSION MODEL

4.1 CURRENT COMPRESSION MODEL

The basic theoretical compression models are presented in Figure 4 and Figure 5. Every compression algorithm or method follows one of these models. Figure 4 depicts the lossless compression model and Figure 5 the more general lossy model (note that the lossless model can be derived from the lossy model by omitting the "Quantization" step). In general, data reduction is achieved by reducing (or ideally, by eliminating) the different types of redundancy. *Pixel mapping* reduces the *interpixel* redundancy. *Quantization* reduces the *psychovisual* redundancy. The final step of *symbol encoding* in the compression model reduces the *coding* redundancy. Usually the pixel mapping and quantization steps do not decrease the size of the data, but instead they decrease the image entropy so that the encoding step can achieve better compression [Rabbani91, Gonzalez93]. Not every compression method does include all steps of this theoretical model, nor can these steps always be clearly identified and distinguished. However, none of the existing compression methods does more than what is presented in this abstract theoretical model.

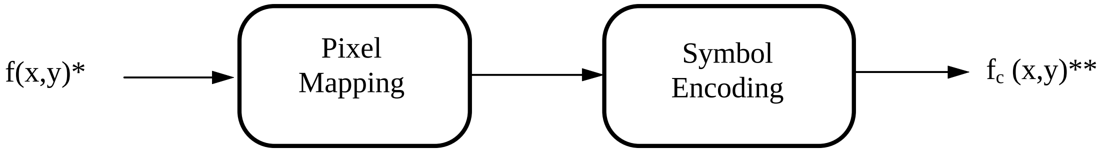


Figure 4: Lossless Compression Model

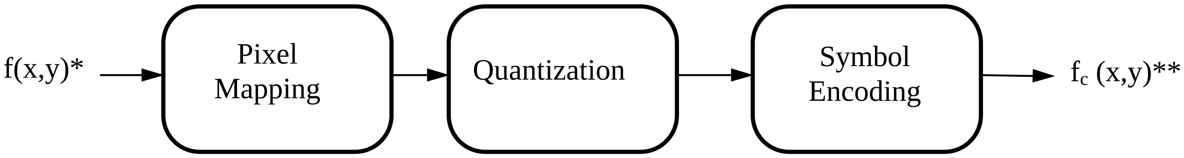


Figure 5: Lossy Compression Model

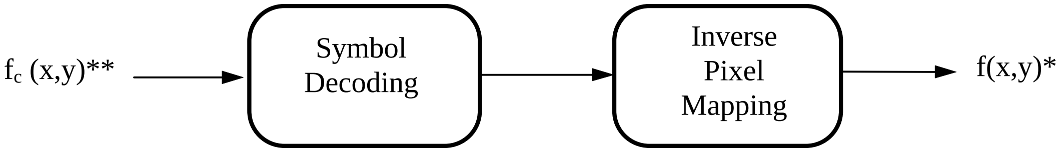


Figure 6: Decompressing an image

* f (x,y) : original image,
** f_c (x,y) : compressed image.

4.2 ENHANCED COMPRESSION MODEL

For individual images, the general compression model described above is sufficient. However, *sets of similar images* contain set redundancy that this general compression model fails to capitalize on. Compression methods based on this model can eliminate only the *intra*-image redundancies, but not the *inter*-image, or *set redundancy*. As it was demonstrated in section 3.2, taking advantage of set redundancy offers the potential to further decrease the entropy of an image, thus achieving better compression rates.

In order to incorporate set redundancy reduction in the theoretical compression model, the *Enhanced Compression Model* is proposed. This is an extension of the basic compression model to include an additional step, the "*set mapping*" step (Figure 7). "Set mapping" reduces the entropy of each image in a set of similar images by taking advantage of set redundancy. This step can be implemented as a "*set mapping transform*", as described in the next section.

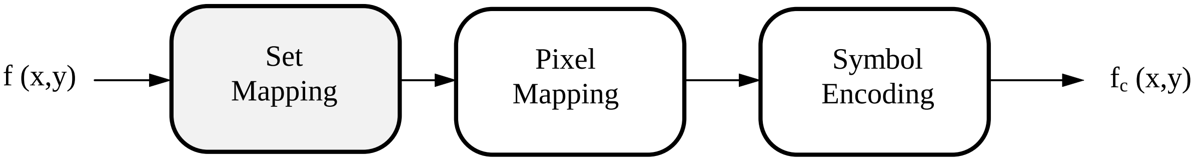


Figure 7: The Enhanced Lossless Compression Model

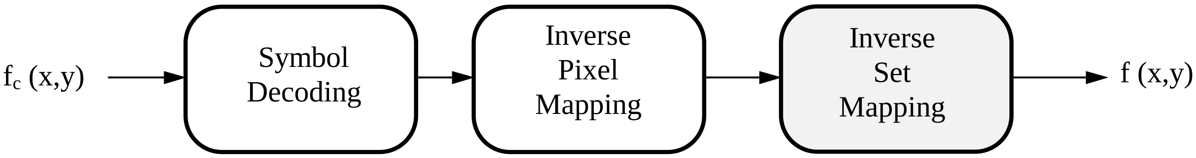


Figure 8: Decompression in the Enhanced Compression Model

4.3 SET MAPPING

The conceptual "set mapping" step in the Enhanced Compression Model can be realized in many different ways. One way is to implement it as an *N*-dimensional transform by translating the origin of the *N*-dimensional coordinate system, where *N* is the number of pixels in a given image. This *N*-dimensional transform can reduce the dynamic range of the pixel values in a set of similar images. This results in fewer bits per pixel required to store these values. The following example clarifies these concepts.

First, assume that every pixel value in an *N*-pixel image represents a coordinate value in the *N*-dimensional space. Consequently, every *N*-pixel image defines a unique point in this space. To visualize these concepts, consider some very small 1x2 "images". Every such image contains only 2 pixels and can be represented by a unique point in the 2-dimensional space. The values of the 2 pixels are used to define the coordinates *X* and *Y*. This is a 2-pixel image:

45	203

and this is the corresponding point in the 2-dimensional space:

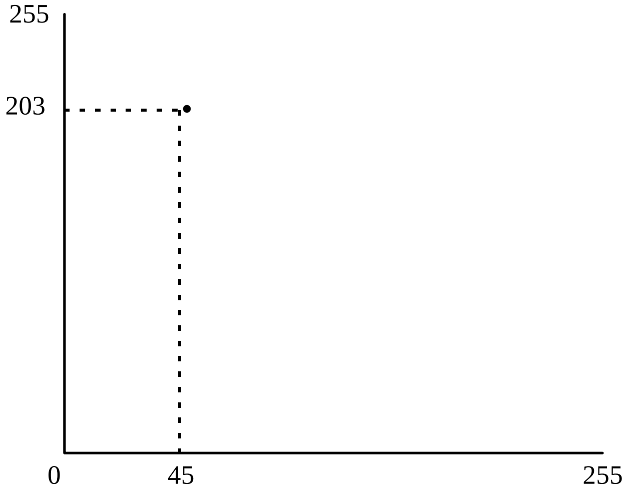


Figure 9: Point defined by a 2-pixel image

A set of random 2-pixel images results in a scatter plot similar to the following:

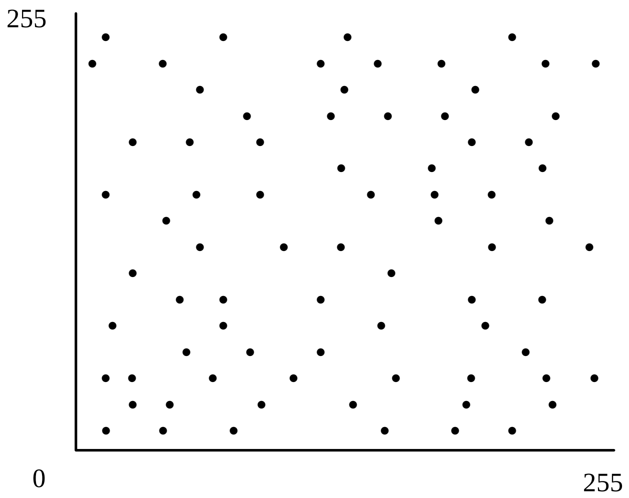


Figure 10: Set of random 2-pixel images

As Figure 10 shows, the range of values is [0-255] for both *X* and *Y* coordinates, therefore 8 bits are needed to represent each value. Now, consider a set of 2-pixel images that are *similar* to each other. In this set suppose that the first pixel's value in every image is in the range [40-60], and the second pixel's value is in the range [135-165]. This is a scatter plot for such a set:

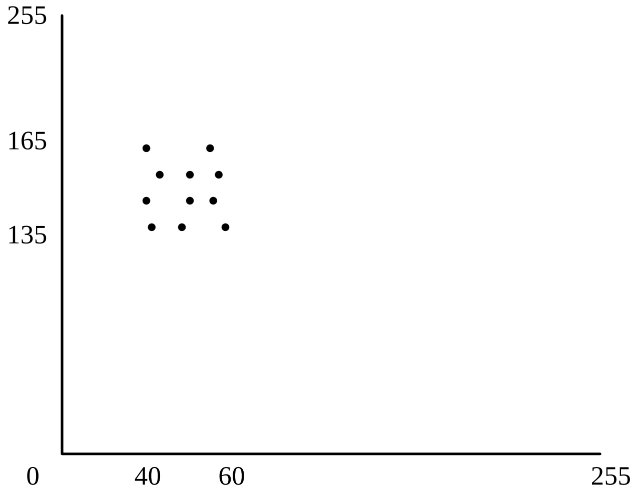


Figure 11: Set of similar 2-pixel images

All the points are clustered together in the same area. In this case, we can translate the coordinate axes so that the points will be closer to the origin. Figure 12 shows this coordinate system transform.

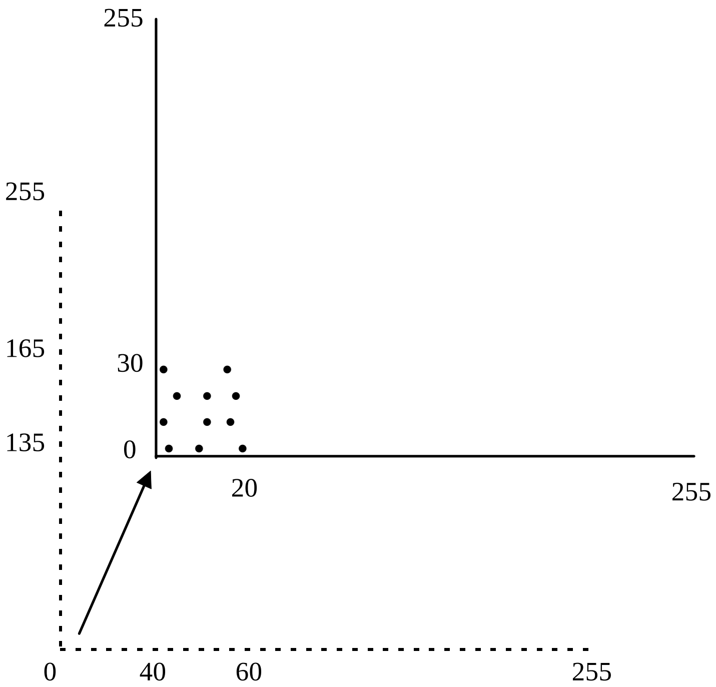


Figure 12: Translating the origin closer to the cluster

After translating the origin, the maximum coordinate values in this cluster of points are X_{max} = 20 and Y_{max} = 30. Therefore, only 5 bits are sufficient to represent these new coordinates, which correspond to the values of the pixels in the transformed images. Of greater importance is the fact that translating the origin caused the variance of the pixel values *inside* every image to decrease: *both* pixels in every image have values in the range [0-30], compared with the original range [40-165]. This represents a decrease of 76% in the variance and results in a reduction of the image entropy. For example, if it were possible to reduce the range of pixel values to [0-7], then the images could be encoded with only 3 bits/pixel. Note that this is *lossless* compression, because it is always possible to perform an inverse axes shift and perfectly recover the original images. In addition, this compression is based solely on the properties of the original *set of images*, and not on the internal properties of the *individual* images. Therefore, additional compression can be achieved by using *any of the existing compression methods* to further compress *individually* every image in this set.

The previous example with 2-pixel images demonstrated how “set mapping” can be implemented as a coordinate origin translation in the 2-dimensional space. The same procedure can be extended to *N*-dimensional space for compressing arbitrarily large images with N pixels each. Note that if the origin is translated at the point where the pixel values of the images become minimum but not negative, then this translation is equivalent to creating the “minimum image” from the set and replacing every image by its difference from this “minimum image”. Analogously, the “maximum image” could be used instead of the “minimum image”. In this case, the origin would be translated at the point where all pixel values become negative, but with minimum absolute values (Figure 13). Finally, the origin could be translated to the centroid of the cluster (Figure 14) which actually corresponds to the “average image” of the image set. In this case, the origin translation is equivalent to creating the “average image” and then replacing every image in the set with its differences from this “average image”.

These ideas about creating and using the “minimum”, “maximum”, or “average” images for set mapping were the starting points for developing practical methods based on the Enhanced Compression Model. These methods are presented in chapter 6. In the next chapter, the requirements and considerations for implementing the Enhanced Compression Model in practice are discussed.

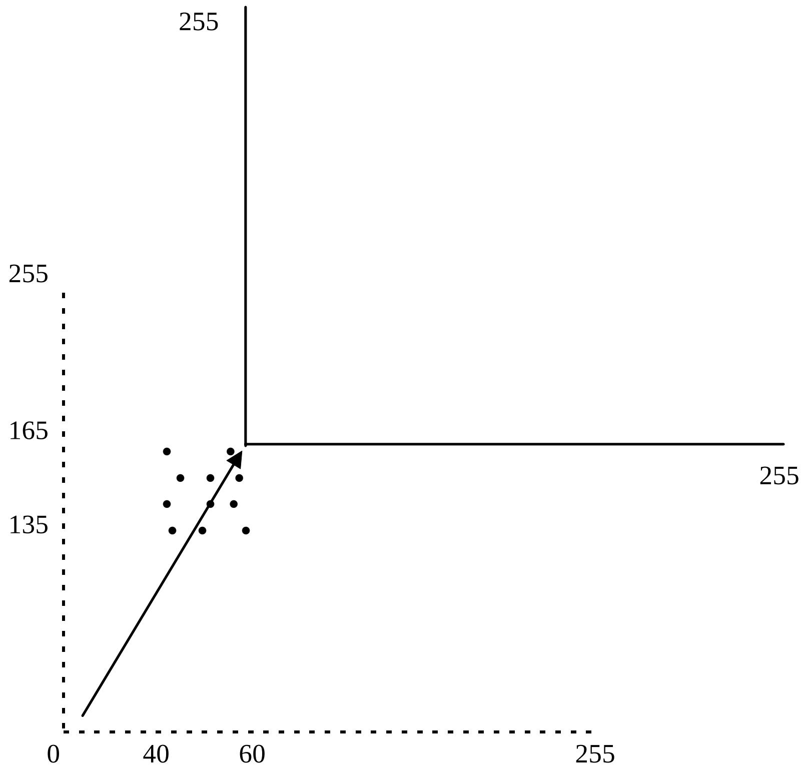


Figure 13: Translating the origin to the “maximum point” of the cluster, or “maximum image”

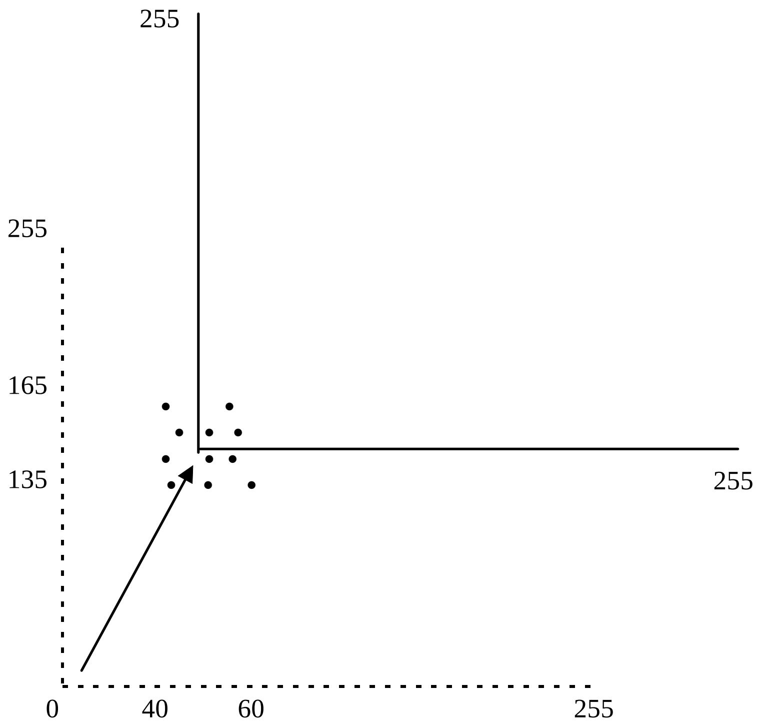


Figure 14: Translating the origin to the centroid point, or “average image”

CHAPTER 5

IMAGE COMPRESSION USING THE ENHANCED COMPRESSION MODEL

The Enhanced Compression Model can be implemented as a two-step procedure (Figure 15). First, the images are decorrelated from the set by extracting the set redundancy; then, the images are compressed by using any compression method. Note that set redundancy extraction is a completely independent step, therefore there are no restrictions on which compression method is used in the second step.

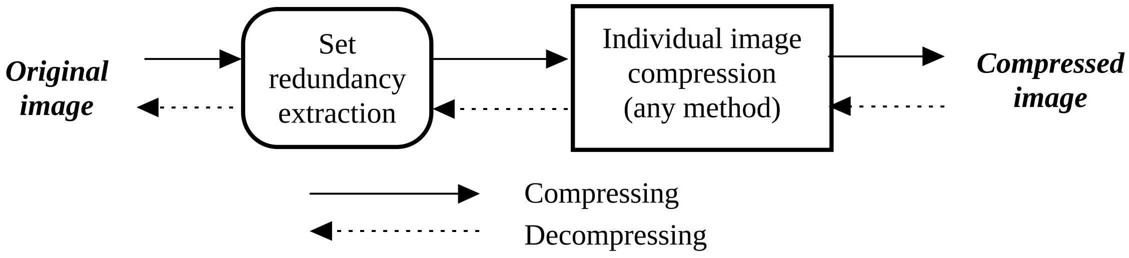

Original image
Set redundancy extraction
Individual image compression (any method)
Compressed image
Compressing
Decompressing

Figure 15: Coupling set redundancy extraction with image compression

Ideally, set redundancy extraction must satisfy the following requirements:

(a) it must reduce (ideally, eliminate) the set redundancy from all images in the set;
(b) it must not be computationally expensive because its computations are in addition to the computations required in the compression of the second step;
(c) it must enable the compression and decompression of individual images from the set without requiring global calculations on the whole set;
(d) it must be lossless.

Currently, there are no methods that have been explicitly designed to extract set redundancy. There are, however, two methods that initially appear to have set redundancy reduction properties. These methods are the Vector Quantization and the Hotelling transform. Nevertheless, a closer examination will show that neither of these two methods can satisfy the requirements for set redundancy extraction.

5.1 VECTOR QUANTIZATION AND SET REDUNDANCY

In Vector Quantization (VQ), the main idea is to replace blocks of pixels with single values. There are many variations of this technique; for a review see [Nasrabadi88]. Generally, VQ starts by constructing a “codebook” with a few common patterns and an index for each pattern. Then, it describes an image using only the indices from this codebook. The codebook is constructed either directly from the image to be compressed (local codebook), or a global codebook can be generated by using several images as a training set [Rabbani91]. The local codebook is the optimal codebook, but it has two disadvantages: (a) generating a local codebook for every image is computationally intensive, and (b) the local codebook must be stored as overhead information together with the image that uses it. A global codebook does not have these disadvantages, and it can be used with good results if the images to be compressed are similar with one another.

VQ compression with global codebook may seem to reduce set redundancy. However, a closer examination reveals that it only eliminates the redundancy in the overhead information (codebook), but not the set redundancy in the images. For example, assume that the first 4 pixel values in every image form an identical vector across all images in the set. This is a typical case of set redundancy. VQ creates an entry in the global dictionary for this vector, and then replaces this vector in all images with its index from the dictionary. Nevertheless, even after this replacement, the set redundancy still remains in the images, in the form of the identical index values that appear at the same position across all images. Furthermore, VQ is typically not lossless because the pixel values usually form similar, but not identical vectors. Therefore, replacing them with one index value discards their unique details. This makes VQ a *lossy* compression method. For these reasons, VQ is not appropriate for set redundancy extraction.

5.2 HOTELLING TRANSFORM AND SET REDUNDANCY

The Hotelling Transform a.k.a. method of Principal Components [Hotelling33] is theoretically optimal in decorrelating the elements of a set of $\boldsymbol{x}$ vectors. From a given set of images, a vector population $\boldsymbol{x}$ can be constructed by grouping the corresponding pixels from all images into vectors (e.g., the vector $\boldsymbol{x_i}$ is formed by grouping the values of pixel position $\mathrm{P_i}$ from all images). If the images are similar, then a correlation exists between the elements of these $\boldsymbol{x}$ vectors. The Hotelling Transform converts the $\boldsymbol{x}$ vectors into $\boldsymbol{y}$ vectors, which have uncorrelated elements. A new set of images can be formed using these $\boldsymbol{y}$ vectors, with the interesting property that most of the variance will be "packed" into only a few of these images. This leads to (lossy) compression if the transformed images with the most information are kept and the others are discarded.

The Hotelling Transform was rediscovered by Kramer and Mathews [Kramer56] and Huang and Schulheiss [Huang63]. Also Karhunen [Karhunen47] and Loeve [Loeve48] developed the Karhunen-Loeve Transform (KLT), which is the implementation of the Hotelling Transform on continuous data. The optimality of KLT was proved by Algazi and Sakrison [Algazi69].

Despite its attractive theoretical decorrelation properties, the Hotelling transform is not appropriate for set redundancy extraction. First, it is a computationally expensive operation because it requires the construction of the covariance matrix for the vector population and the calculation of the eigenvectors for this matrix. The large amount of calculations usually prohibit the use of the Hotelling transform in practice. Moreover, the covariance matrix may be singular, making impossible to uniquely define some of its eigenvectors. In addition to the mathematical disadvantages, the Hotelling transform cannot compress or decompress single images from the set without performing global calculations involving all images in the set. For all these reasons, the Hotelling transform cannot be used in practice for set redundancy extraction.

5.3 CONCLUSION

The lack of practical methods for set redundancy extraction would limit the usefulness of the Enhanced Compression Model. Therefore, one of the goals of this research was to develop practical set redundancy extraction methods. These methods are employed by the Enhanced Compression Model to improve compression and are referred to as Set Redundancy Compression, or SRC methods. In the next chapter, three SRC methods will be presented. These are the Min-Max Differential method (MMD), the Min-Max Predictive (MMP) method, and the Centroid method. These methods satisfy all the requirements mentioned in this chapter for practical set redundancy extraction. Specifically, they do not require many calculations, they are easy to implement, they can compress or decompress individual images from a set of images, and they are lossless.

CHAPTER 6

SET REDUNDANCY COMPRESSION (SRC) METHODS

6.1 THE "MIN-MAX DIFFERENTIAL" (MMD) METHOD

In chapter 4, a theoretical study of the set mapping operation produced the idea of creating and using "minimum" and "maximum" images to reduce set redundancy in a set of similar images. To create the "minimum image" (Min), the pixel values across all images are compared, and for each pixel position the smallest value is chosen. Similarly, the "maximum image" (Max) is created by selecting the largest pixel value for each pixel position. Then, set redundancy can be reduced by replacing every image in the set with its differences from either the Min or the Max image.

Figure 16 presents graphically this operation. The abscissa represents the pixel positions and the ordinate the pixel values. Every curve describes an image. The curves for the Min, the Max, and a random image from the set are depicted. In this case, the differences from the Min image are used. The difference values that replace the original image values are shown as dotted lines.

Note that for all pixel positions, the difference values are smaller than the original pixel values. However, there are regions where the differences would be even smaller if the Max image was used instead of the Min image (for example, in Figure 16, the region between pixels P_6 to P_{10}). In general, it is not expected that all pixels in an image will be consistently close to either Min or Max; rather, some regions of an image will be closer to Min, while other regions will be closer to Max. Clearly, a method flexible enough to switch between the Min and Max curves wherever appropriate would produce better results.

This observation led to the development of the Min-Max Differential (MMD) method. MMD uses both Min and Max curves, and for every pixel position it finds and stores the smallest difference value (Figure 17). To synchronize encoding and decoding, the encoder uses consistently one of the two curves until it finds a difference value larger than (max-min) / 2. In that case, it encodes this value and switches to the other curve. The decoder follows the same rule; when it finds a difference larger than (max-min) / 2 it also switches to the other curve.

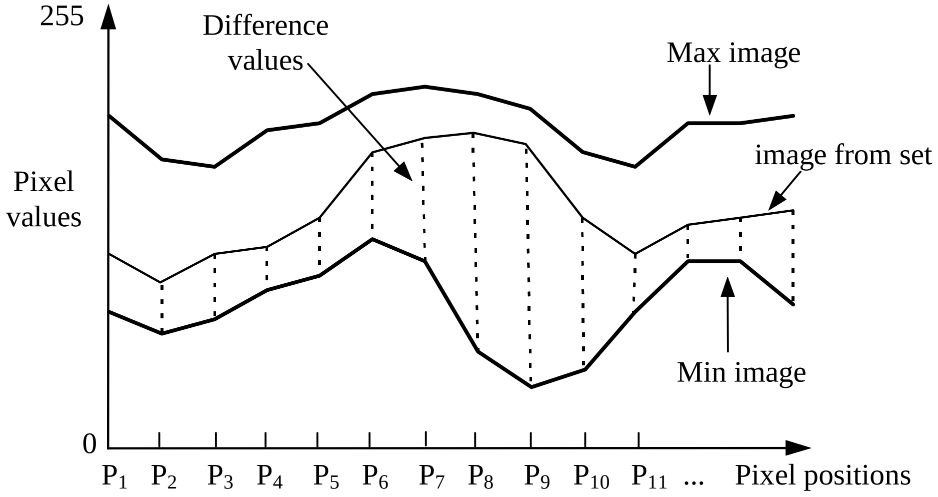


Figure 16: Finding the differences from the "minimum image" (Min)

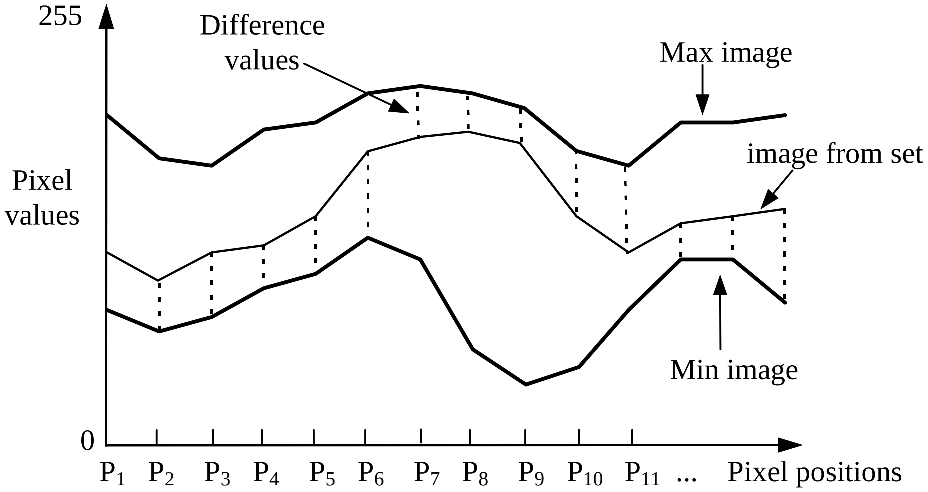


Figure 17: The Min-Max Differential method

6.2 THE "MIN-MAX PREDICTIVE" (MMP) METHOD

The MMD method presented in the previous section is a very simple and useful method to reduce set redundancy. However, its simplicity does not allow 100% use of the available resources. These resources are the Min and Max images that are available during the encoding and decoding operations. Note that MMD uses these images *piecewisely*; in other words, for every pixel position it uses *either* the Min *or* the Max image, but not *both*. In this way, it uses only half of the available statistical information for a given pixel position. The Min-Max Predictive (MMP) is a more complicated, but also a more powerful method that uses *both* Min and Max images.

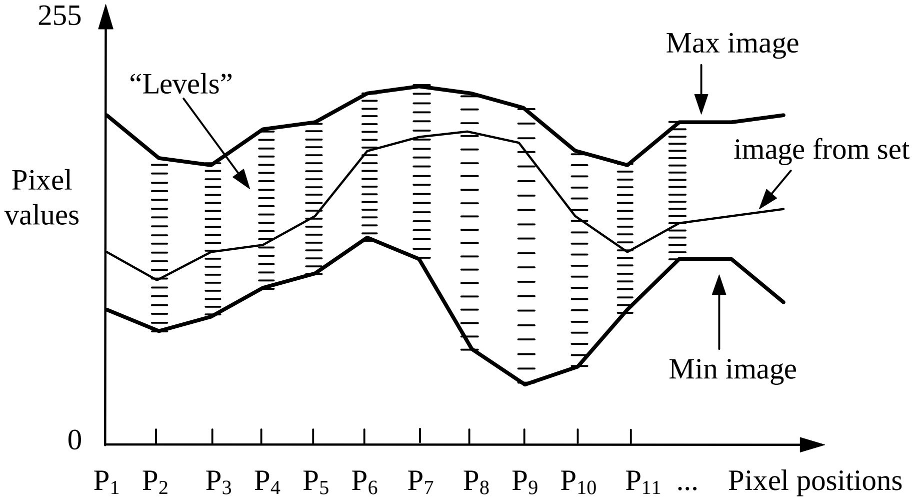

255
Max image
"Levels"
image from set
Pixel values
Min image
0
P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9 P_{10} P_{11} ...
Pixel positions

Figure 18: The Min-Max Predictive method (N=20)

For every pixel position P_i, the Min image gives the minimum value min_i across all images in the set. Similarly, the Max image provides the maximum value max_i. These two values are the limits for the range of possible values that pixel P_i may assume. Note that neighboring pixels tend to fall in approximately the same area within this range (Figure 18). By dividing the range between the minimum and maximum values into N levels, we can represent the position of every pixel between its corresponding minimum and maximum values as a "level" L_i:

$$L_i = N\left(\frac{\text{value}(P_i) - \min_i}{\max_i - \min_i}\right).$$

Neighboring pixels usually have approximately the same “level”, even though their actual values may differ. For example, consider the following values:

Maximum value	192	205	211	197	183	190	199	204	212	223
Pixel value	71	78	89	85	69	80	87	95	106	110
Minimum value	14	18	30	30	13	27	33	41	54	54

For every pixel position, the range between its maximum and minimum values can be divided into N=256 “levels”, where 0 represents the minimum value, and 255 the maximum value. According to this definition, these pixel values correspond to the following “levels”:

Pixel value	71	78	89	85	69	80	87	95	106	110
Level	82	82	83	84	84	83	83	84	84	85

As this example demonstrates, the “level” values have smaller variation than the pixel values. Therefore the “levels” are better predictors for the next pixel value than the pixel values themselves. This observation led to the development of the MMP method. The MMP method predicts the value of a pixel P_i by using the “level” information from its previous pixel $\mathrm{P}_{\mathrm{i-1}}$. Pixel $\mathrm{P}_{\mathrm{i-1}}$ and its $min_{\mathrm{i-1}}$ and $max_{\mathrm{i-1}}$ are used to determine its level $L_{\mathrm{i-1}}$. This can be used directly as a predictor for the level of the current pixel ($L_\mathrm{i} = L_{\mathrm{i-1}}$). A variation with slightly better results is to set $L_\mathrm{i} = (L_\mathrm{upper} + L_\mathrm{left}) / 2$ where L_upper is the level of the upper neighboring pixel and L_left is the level of the left neighbor. Finally, another variation is to set $L_\mathrm{i} = L_\mathrm{upper} + L_\mathrm{left} - L_\mathrm{upperleft}$. These three variations of the MMP method are referred to as MMP1, MMP2, and MMP3, respectively. After calculating L_i, a prediction for the value of pixel P_i can be found as follows:

$$\text{prediction}(P_i) = \min{}_i + \frac{L_i}{N}\left(\max{}_i - \min{}_i\right) .$$

The difference between this predicted value and the original value is stored in the image replacing the original value. This concludes the encoding process. To recover the original image from its “difference image”, the decoding process follows the same steps to calculate the predicted values, and then it adds the difference values obtained from the “difference image”.

6.3 THE "CENTROID" METHOD

In general, predictive methods are very useful for image compression. These methods create a prediction for the value of every pixel; then the encoder stores only the error between this prediction and the true pixel value. If the prediction scheme is good, then the errors are very small, having a Laplacian distribution with most of the values very close to zero. The MMP method developed in the previous section is a predictive method that uses the "minimum" and "maximum" images for predicting every pixel's value. Another predictive method would be to use the "average image" from the set of similar images. A simple scheme for predicting the pixel value at position i in image j is:

$$F_{i,j} = m_i \tag{Eq. 2}$$

where m_i is the average value from position i across all images. This is a very simple prediction scheme that results from intuition rather than mathematical development. However, a formal analysis is often necessary to produce more sophisticated and more accurate models. In this section the mathematical foundation for a better prediction scheme that uses more effectively the "average image" will be developed.

Let us assume we have a set of K images, with N pixels per image. We define $x_{i,j}$ to be the pixel value of pixel $\mathrm{P_i}$ from image j. For natural images, a model that can describe this value is:

$$x_{i,j} = x_{i-1,j} + r_{i,j} + s_{i,j} \tag{Eq. 3}$$

where $x_{i\text{-}1,j}$ is the value of the previous pixel, $r_{i,j}$ is a random error, and $s_{i,j}$ is the difference of the two pixels due to the local change in intensity values (slope). Taking the sum of (Eq. 3) for pixel position i across all K images and dividing by K, results in:

$$\frac{\sum_{j=1}^{K} x_{i,j}}{K} = \frac{\sum_{j=1}^{K} x_{i-1,j}}{K} + \frac{\sum_{j=1}^{K} r_{i,j}}{K} + \frac{\sum_{j=1}^{K} s_{i,j}}{K} \ . \tag{Eq. 4}$$

Since $r_{i,j}$ is a random variable, we have

$$\sum_{j=1}^{K} r_{i,j} = 0 \ .$$

Also, the average value m_i of pixel i across all images is defined as:

$$m_i = \frac{\sum_{j=1}^{K} x_{i,j}}{K} \ .$$

Therefore (Eq. 4) becomes:

$$m_i = m_{i-1} + \frac{\sum_{j=1}^{K} s_{i,j}}{K} \ . \qquad \text{(Eq. 5)}$$

Subtracting (Eq. 5) from (Eq. 3) yields:

$$x_{i,j} - m_i = x_{i-1,j} + r_{i,j} + s_{i,j} - m_{i-1} - \frac{\sum_{j=1}^{K} s_{i,j}}{K}$$

thus,

$$x_{i,j} = m_i + \left(x_{i-1,j} - m_{i-1} + r_{i,j} + s_{i,j} - \frac{\sum_{j=1}^{K} s_{i,j}}{K} \right)$$

or,

$$x_{i,j} = m_i + \varepsilon_{i,j} \qquad \text{(Eq. 6)}$$

where,

$$\varepsilon_{i,j} = x_{i-1,j} - m_{i-1} + r_{i,j} + s_{i,j} - \frac{\sum_{j=1}^{K} s_{i,j}}{K} \ . \qquad \text{(Eq. 7)}$$

The term $\varepsilon_{i,j}$ can be considered as an “error term” that denotes the difference of pixel value $x_{i,j}$ from the mean value m_i at pixel position i. From (Eq. 6) we have:

$$x_{i-1} - m_{i-1} = \varepsilon_{i-1,j} \, . \qquad \text{(Eq. 8)}$$

Because the images are similar, the slope difference $s_{i,j}$ at pixel position i of image j will be very close to the average slope difference at the same pixel position i across all images. Therefore,

$$s_{i,j} - \frac{\sum_{j=1}^{K} s_{i,j}}{K} = 0 \ . \qquad \text{(Eq. 9)}$$

By replacing (Eq. 8) and (Eq. 9) at (Eq. 7) we get:

$$\varepsilon_{i,j} = \varepsilon_{i-1,j} + r_{i,j} \ . \qquad \text{(Eq. 10)}$$

To summarize, from (Eq. 3) and with the assumption of similarity across images we have derived equations (Eq. 6) and (Eq. 10) which define the following model for a pixel value in a set of similar images:

$$\begin{aligned} x_{i,j} &= m_i + \varepsilon_{i,j} \\ \varepsilon_{i,j} &= \varepsilon_{i-1,j} + r_{i,j} \end{aligned} \qquad \text{(Eq. 11)}$$

In this model, the error term is *serially correlated* or *autocorrelated.* Forecasting with autocorrelated error terms has been studied extensively in statistical theory [Neter89]. Linear regression with one independent variable and first-order autoregressive error is defined as:

$$\begin{aligned} x_{i,j} &= \beta_0 + \beta_1 \cdot m_i + \varepsilon_{i,j} \\ \varepsilon_{i,j} &= \rho \cdot \varepsilon_{i-1,j} + r_{i,j} \end{aligned} \qquad \text{(Eq. 12)}$$

where $x_{i,j}$ is the dependent variable, m_i is the independent variable, β_0 and β_1 are the regression parameters, ρ is the autocorrelation parameter, and $r_{i,j}$ is an independent and normally distributed random error term with zero mean value. Clearly, the model defined by (Eq. 11) is the same as the linear regression model defined by (Eq. 12), with:

$$\beta_0 = 0 \ , \quad \beta_1 = 1.0 \ , \quad \text{and} \quad \rho = 1.0 \qquad \text{(Eq. 13)}$$

In statistical theory it is proven that a good prediction scheme for the model of (Eq. 12) is [Neter89]:

$$F_{i+1,j} = \beta_0 + \beta_1 \cdot m_{i+1} + \rho \cdot e_{i,j}$$

where $F_{i+1,j}$ is the forecast for the (unknown) value $x_{i+1,j}$, and $e_{i,j}$ is defined as:

$$e_{i,j} = x_{i,j} - \left(\beta_0 + \beta_1 \cdot m_i\right) .$$

Using (Eq. 13) these equations are simplified to:

$$F_{i+1,j} = m_{i+1} + e_{i,j}$$
$$e_{i,j} = x_{i,j} - m_i$$

These two equations define a model for predicting the value of a pixel using the average value for that pixel position plus a correction term. The experimental results that are presented in chapter 7 show that this model is more accurate and provides better compression than the simpler model of (Eq. 2).

6.4 The Use of Robust Statistics

The problem with the average, minimum, or maximum is their sensitivity to outliers. For example, suppose that in a certain data distribution all values are within the range 40-60, except one outlier with the value 190. In this case, the maximum will be 190, even though it is not representative of the distribution. Furthermore, the average value will also be skewed toward this outlier. Statistics sensitive to outliers can potentially degrade the performance of SRC methods. To avoid this, *robust statistics* can be used.

One statistical measure that is more robust to the presence of outliers is the median value. Median is defined as the central value in a sorted data set (or, the mean of the two central values). From a given set of images, the “median image” can be constructed instead of the “average image”. Then, a statistically robust version of the Centroid method would use this “median image” instead of the “average image”.

To increase the robustness of the minimum and maximum, a certain portion of values at the tail regions of the distribution can be ignored. The rationale is that outliers by definition lie in the tail regions of the distribution, therefore by ignoring these tails the outliers can be avoided. For example, the maximum value in a dataset with 100 points and 5% “robustness shield” would be the 6th largest value. In this way, “robust maximum” and “robust minimum” images can be created, which can be used to increase the statistical robustness of the MMD and MMP methods.

6.5 Other Methods

Other approaches were also investigated as potential methods for set redundancy extraction. However, it was found that there were drawbacks to their practical implementation, therefore they were not included in the group of SRC methods developed in this research. These approaches and their practical disadvantages are briefly discussed next.

6.5.1 The "Block-Adaptive Huffman" Method

The basic idea in this method was to divide every image into non-overlapping blocks and use Huffman encoding with a different table for each block category. These local Huffman tables would be constructed by pooling together the data from the corresponding blocks across all images. The rationale was that the use of specialized Huffman tables would improve compression. However, preliminary experimental results were discouraging. It seems that there is not enough skewness in the data for each block category to benefit from the use of local Huffman tables. Therefore, this idea was not used in the development of the SRC methods.

6.5.2 The "Cluster-Adaptive Huffman" Method

Another idea was to create artificial skewness in the data by segmenting the images into regions. This segmentation process would define regions of pixels with similar values, so that the data distribution inside every region would become highly skewed. Then, region-based Huffman tables could be created by pooling the data from corresponding regions across all images. Theoretically, this method should result in better compression. However, there are drawbacks to its implementation. One drawback is the difficulty with segmenting the images. Image segmentation is computationally expensive, and the algorithms are complicated. As a result, this method would violate the requirements discussed in chapter 5 for simplicity and low computational cost. Moreover, storing the region boundaries for every image would increase the size of the datasets. Therefore, this method was not included in the SRC methods.

6.5.3 The "Video Sequence" Method

The frames in a short video sequence represent an *ordered* set of similar images. There are several existing video compression methods (see Appendix B). These methods take advantage of the well-defined sequence of frames to reduce temporal redundancy. In a set of similar images, there is no temporal redundancy, hence video compression methods cannot be used effectively.

Nevertheless, it might be possible to convert part of set redundancy into temporal redundancy. The idea is to find the proper ordering for the images that would minimize the differences between every image from the next. A set of images with such ordering would resemble frames in a video sequence, therefore, video compression algorithms could be used.

This approach however has a number of drawbacks. First, finding the optimal sequence is similar to searching for the optimal path in the Traveling Salesman Problem; it is a combinatorial problem. Second, an image can not be decompressed without decompressing a number of other images, therefore the requirement (see chapter 5) for decompressing individual images would be violated. Finally, most of the video compression methods are lossy, therefore they are not suitable for this type of compression. For all these reasons, neither this approach was used in the SRC methods.

6.6 SUMMARY

In this chapter three SRC methods were presented: the Min-Max Differential method, the Min-Max Predictive method, and the Centroid method. These methods reduce set redundancy in a set of similar images by creating and using statistical images from the set, such as the “minimum”, “maximum”, or “average” image. The use of robust statistics was discussed as a way to reduce the effects of outliers. Three other approaches were also examined as possible ways to reduce set redundancy: the “Block-Adaptive Huffman”, the “Cluster-Adaptive Huffman”, and the “Video Sequence” method. However, it was found that these approaches do not satisfy the requirements for practical set redundancy extraction discussed in chapter 5. Therefore, the conclusion was that these approaches should not be included in the group of SRC methods developed in this research.

CHAPTER 7

APPLICATION OF THE ENHANCED COMPRESSION MODEL ON MEDICAL IMAGES

Medical imaging is a field that changes rapidly due to digital technologies. The development of an all-digital radiologic environment is expected to offer great benefits to medical centers. However, it will also result in overwhelming amounts of digital imaging data, that will impose exceedingly high data storage and communication requirements.

In this chapter the importance of image compression in medical imaging is discussed, and the application of the Enhanced Compression Model in a medical image database is presented. According to the Enhanced Compression Model, the performance of standard compression methods can be improved when they are combined with SRC methods. The results presented in this chapter verify the validity and the practical importance of the Enhanced Compression Model.

7.1 THE DIGITAL RADIOLOGIC ENVIRONMENT

During the last half century, digital imaging technologies have revolutionized the field of medical imaging. The radiologic environment has become increasingly digital, leading to the anticipation of the “filmless hospital”. Picture Archival and Communication Systems (PACS) are the integrated electronic systems that will support this digital environment by aiding in the collection, storage, and communication of medical digital images.

The benefits and problems associated with an all-digital radiologic environment have been shown in several studies [Kositpaiboon89, Chimiak92, Chipman92, Cox92, Dwyer92, Huang92, Keizers92, Orozco92]. Some of the most important benefits emanate from the replacement of film libraries by digital image archiving. Films stored in film libraries are often misfiled, damaged, lost, and occupy valuable space. Retrieving an image from a film library is time consuming because a person must physically search and retrieve the requested film. In addition, an image on film is no longer available for digital processing such as window and level manipulation and it cannot simultaneously exist in more than one location. In contrast, archiving of digital images has none of the above mentioned problems. Digital images are always available for digital processing, can be transmitted through computer networks, and can be accessed by more than one person at the same time. Finally, digital archiving provides response times that are far better than film library systems and operations like search and retrieval can become fully automated.

Nevertheless, there are also some problems that impede the wide acceptance of the all-digital radiologic environment. One of these problems is the lack of economical displays that have both the required brightness and high resolution. Currently, radiological examinations are performed with films and viewboxes even though the technology exists for manufacturing high-quality computer displays. If vendors are convinced about the size of this potential market, then inexpensive customized displays will soon replace viewboxes. Another problem in digital radiology is the lack of definitive standards. There is a large number of proprietary data formats, communication protocols, and system interfaces. This cacophony impedes the integration and communication of digital medical systems. It is believed that the situation will improve as the field matures and standards like the ACR/NEMA DICOM* are adopted. However, the major problem in implementing an all-digital radiologic environment is the need to store, communicate, and manage huge amounts of data. Studies have shown that the radiology department of a large hospital can produce more than 20 terabits of image data per year [Wong92, Keizers92]. The large data requirements stem from the high resolutions needed for medical images and the large number of images required for each examination. Table 1 presents the storage requirements for the main digital techniques used in medical imaging [Wong95] which are Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), Single Photon Emission Computerized Tomography (SPECT), Ultrasonography (US), Digital Subtraction Angiography (DSA), and Digital Flurography (DF).

Table 1: Amount of data produced in medical imaging

Modality	CT	MRI	PET	SPECT	US	DSA	DF
Image Dimension (one side)	512	256	128	128	512	1024	1024
Bits per pixel	12	12	16	8 or 16	6	8	8
Avg. # images per exam	30	50	62	50	36	20	15
Avg. Mbytes per exam	12	5	2	1 or 1.5	7	21	16

* DICOM (Digital Imaging and Communication in Medicine) is a standard being jointly developed by the American College of Radiology (ACR) and the National Electrical Manufacturing Association (NEMA).

These numbers increase as more dimensions are added to the techniques. For example, new and faster MRI techniques have introduced a temporal dimension to MRI that significantly increases the size of the datasets. In addition to the inherently digital modalities, digital data are also being produced by digitizing X-ray images. A digitized chest X-ray requires 4K x 4K x 12 bits, or 24 Mbytes per film image.

Overall, the amount of digital radiologic data generated every year in the USA alone is on the order of petabytes (10^{15}) and is increasing rapidly [Wong95]. This strains the capabilities of digital storage systems, and imposes exceedingly high requirements on the bandwidth of communication networks. Digital image compression can address these problems by reducing the data storage and transmission requirements. Many compression methods have been developed and have been evaluated for the medical environment. These compression methods usually reduce the size of the data 2 to 3 times with no information loss and more than 10 times with some information loss. In the next section, a brief review of methods used in radiologic image compression will be presented.

7.2 IMAGE COMPRESSION IN MEDICAL IMAGING

Lossy image compression research in medical imaging has traditionally concentrated on transform methods, especially the 2-D DCT [Lo85, Elnahas86, Chan89, Ishigaki90]. Two promising alternatives to DCT are Vector Quantization [Riskin90, Cosman93] and Wavelet Transform [Saipetch95]. Other lossy compression methods include DPCM with scalar quantization [Chen94], Lapped Orthogonal Transform, Subband Coding, and Quadtrees [Wong95].

Despite the higher compression ratios of lossy compression methods, their use in medical imaging is limited because of concerns on losing image details. Even when the compression is visually lossless, an unsuccessful diagnosis from an image that has lost some information may lead to legal implications [Wong95]. Another reason for avoiding lossy compression is the potential development of computer-aided diagnosis techniques. Computerized analysis of an image can use even the smallest details (e.g., very smooth variations in pixel intensities) which are often invisible to the eye. Compression methods should not lose any of these potentially important image details.

For these reasons, many studies in medical image compression have focused on lossless methods, such as Run-Length Encoding [Rhodes85], HINT [Roos93], Multiplicative Autoregression [Das93], Differential and Huffman encoding [Bramble89], Laplacian pyramids, Bit-Plane Decomposition, and the use of contexts [Ramabadran92]. A good review of various methods is Wong *et.al.* [Wong95]. Also there are several studies that compare lossless compression methods for use in radiology, for example [Roos88, Roos91, Kuduvalli92].

7.3 SET REDUNDANCY IN MEDICAL IMAGE DATABASES

Medical imaging is an area in which SRC compression methods can be used very effectively. The reason is that medical images always depict parts of the same subject: the human body. Therefore, medical image databases store thousands of images that are very similar to one another. Moreover, these images can be classified into categories, according to the modality that produced them and the specific organ or part of the body they depict (for example, chest X-rays, brain axial MRI scans, liver CT scans, etc). This classification results in sets of images with high similarity, that contain large amount of set redundancy. According to the Enhanced Compression Model, this set redundancy can be used to improve compression.

Improved compression of medical images results in more efficient storage and faster transmission of images. Efficient storage is important for large medical centers that produce and store thousands of new images every year. Faster transmission is essential for Telemedicine applications.

To demonstrate the application of SRC methods in medical imaging, a test database of CT and MRI brain scans was used. According to the Enhanced Compression Model, SRC methods were used to extract set redundancy from the images before any compression method compresses them. The combination of SRC methods with three standard compression methods was tested: Huffman encoding, Arithmetic coding, and Lempel-Ziv compression. The next sections present the experimental results.

7.4 EXPERIMENTAL RESULTS

7.4.1 The Test Database

A test database of 51 CT brain images and 57 MR brain images was used to evaluate the performance of the SRC methods (the images are presented in Appendix D). All images were obtained at M.D. Anderson Cancer Center in Houston, Texas. The resolution was 512x512 for the CT and 256x256 for the MR. All images were gray-level, and were scaled to 8 bits/pixel. They were randomly selected from patients of both sexes, various ages, and a variety of pathological conditions. From this test database, two image sets with high similarity were formed, one set from the CT and the other from the MR images. These two sets contained 10 images each; this set size was chosen to ensure high similarity among the images. The remaining images were considered to belong in other sets. In this small test database, the size of those other sets was not large enough for testing purposes. However, a real database containing hundreds or thousands of images can be partitioned in a number of sets with reasonably large sizes; in that case SRC methods can be efficiently implemented in every set.

Note that creating sets of similar images in an image database is equivalent to the cluster generation problem. This problem can be easily solved when external information that can aid image classification is available. However, the problem is more challenging when no external information exists, and the images are to be clustered based only on pixel data. In this research, no external information was used.

When image clustering is based on pixel data, a "distance metric" is used as a measure of similarity between two images. There are several distance metrics that can be used for image clustering, for example, correlation coefficient, mean-squared error, sum of absolute differences, etc. In this research, the average absolute differences between the pixel values of images was used as a distance metric. The reason is that this metric involves only additions and subtractions of pixel values, therefore can be calculated fast, and it is also robust to outliers.

The next section presents the clustering scheme that was used to create the sets of similar images from the test database. The goal was to demonstrate how the clustering problem can be solved when there is no external information available to aid the image classification.

7.4.2 Image Clustering

The problem of forming sets of similar images from an image database is known as the *classification,* or *cluster generation* problem. When there is external information about the content of the images, then the problem has a trivial solution. In the general case however, classification is based solely on some image distance metric and it becomes a combinatorial problem with exponential time complexity. To test the SRC methods, a variation of the classification problem was used, according to which only one cluster of a specific size is to be formed. This cluster corresponds to the set of images with the greatest similarity. Even in this simple case, the time complexity is prohibitively large. To find the best cluster of size k, an exhaustive search would require

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

cluster formations and evaluations, where n is the total number of images. For a test database of 50 images, searching for the best 10-image set would require 10,272,278,170 cluster evaluations. Even if a computer were powerful enough to perform 100 cluster evaluations per second, it would take more than 3 years to find the solution! Clearly, exhaustive search is not practical to solve this problem. Therefore, ad hoc or heuristic optimization methods are usually used in practice. In this research, *genetic algorithms* were used to find an acceptable solution.

Genetic algorithms is an optimization technique that was formulated during the early years of the 1970's by John Holland [Holland75]. This technique is useful for finding the optimal or near optimal solutions for combinatorial optimization problems. It is based on the assumption that simulating an evolutionary process in a population of potential solutions can eventually "evolve" good solutions. Genetic algorithms start with an initial population of random solutions that form the first generation. Then "*crossover*" and "*mutations*" are used to create the next generations. Crossover combines parameters from current solutions to create new ones while mutations modify a few of these parameters. For every generation, the best solutions (according to their "*fitness value*") are selected to form the next generation. This process is repeated until satisfactory solutions evolve. Even though it is not guaranteed that the optimal solution will be found, genetic algorithms is a technique that can rapidly produce solutions very close to the optimal. The pseudo code for the genetic algorithms is the following:

```
Create_Random_Initial_Population
For Generation=1 To Total_Generations Do
   Begin
      Do_Random_Crossover
      Do_Random_Mutations
      Select_Best_Solutions_For_Next_Generation
   End
Output_Best_Solution_Found
```

For the purpose of image clustering, every solution represents a potential cluster (or, set) of images. In this case, "crossover" of solutions is defined as the process of exchanging some of the images between two clusters. For example, assume that the construction of the best 5-image set is sought. If {3, 8, 10, 15, 43} and {7, 19, 21, 35, 38} were two solutions (these numbers correspond to image indices), then "crossover" would randomly exchange parts of these cluster configurations. A possible result may be {3, 8, 21, 35, 38} and {7, 19, 10, 15, 43}.

In this research, the set size for clustering was chosen to be 10. Generally, the set size is directly related to the performance of the SRC compression methods. Smaller set sizes result in higher overall similarity among the images of the set, therefore, better compression. However, the overhead of storing the "average", "minimum", or "maximum" images is relatively higher for smaller set sizes. The optimal set size also depends on the total number of images in the database. The more images a database contains, the larger the sets that can be created. For the test database used in this research, the size of 10 was considered to be an appropriate size for ensuring high similarity among the images included in the set.

The "fitness value" of each set was defined as the inverse of the summation of differences between its images. In this way, larger fitness values imply smaller total differences and, therefore, higher similarity among the images. In order to avoid repetitive calculations of image differences, a table with the differences between all image pairs was created before implementing the genetic algorithms.

For the test database used in the experiments, genetic algorithms produced good 10-image sets with less than 30 seconds of computational time, using 50 generations and a population of 1000 solutions. The two resultant sets of images were the following:

{CT08, CT15, CT19, CT21, CT22, CT24, CT25, CT37, CT43, CT50}, and
{MR21, MR27, MR28, MR29, MR38, MR39, MR44, MR50, MR52, MR55}.

7.4.3 Image Registration

To facilitate the SRC compression methods, the original images were registered into a standard position and size. This was done with a semi-automatic procedure, in which three landmark points were identified in every image and then the images were rotated, translated, and scaled so that these three landmark points would coincide with standard positions. This registration is useful in creating the “minimum”, “maximum”, and “average” images. Note, however, that there is no need to store the registered versions of the original images. Once the registration parameters (rotation, translation, and scaling) for every image are known, these parameters can be used to perform the inverse registration on the “minimum”, “maximum” and “average” images, instead of using the registered versions of the original images. Consequently, the original images can be used without change.

7.4.4 Experimental Procedure

As explained in chapter 5, the Enhanced Compression Model suggests combining the use of SRC methods with individual-image compression methods. The theoretical analysis of chapter 4 has shown that pre-processing a set of images with a SRC method increases the compressibility of the images, i.e., standard compression methods then can be used with improved results.

To support the theoretical analysis with experimental results, all SRC methods were tested in combination with three well-known standard compression techniques: Huffman encoding, Arithmetic coding, and Lempel-Ziv compression (a brief description of these techniques was presented in chapter 2). The first two are entropy-based techniques; the third is a dictionary-based compression scheme. Huffman encoding is one of the oldest and most widely used compression methods. Arithmetic coding is not as popular as Huffman, mainly because it is slower and more difficult to implement. However, it outperforms Huffman method in compression ratios. Lempel-Ziv compression is based on a different approach than these other two methods. There are many versions of Lempel-Ziv compression; in this research the LZW version [Welch84] was used.

All experiments were performed under the Unix operating system which includes standard implementations of Huffman and LZW compression methods (commands “pack” and “compress”, respectively). For Arithmetic coding, the publicly available software based on [Moffat95] was used.

Every SRC method was tested by combining it with each of these three compression methods. The experimental results are presented in the next two sections. The first section presents the results from compressing the CT images, and the second section the results from compressing the MR images.

7.4.5 Application of SRC Methods on CT Images

The set of CT images used in the experiments contained axial CT brain scans which depicted a horizontal slice of the brain at the eye-level (see Appendix D). From this set, the “average”, “minimum”, and “maximum” images were created to be used in the MMD, MMP and Centroid methods. Figure 19, Figure 20, and Figure 21 respectively present these three images.

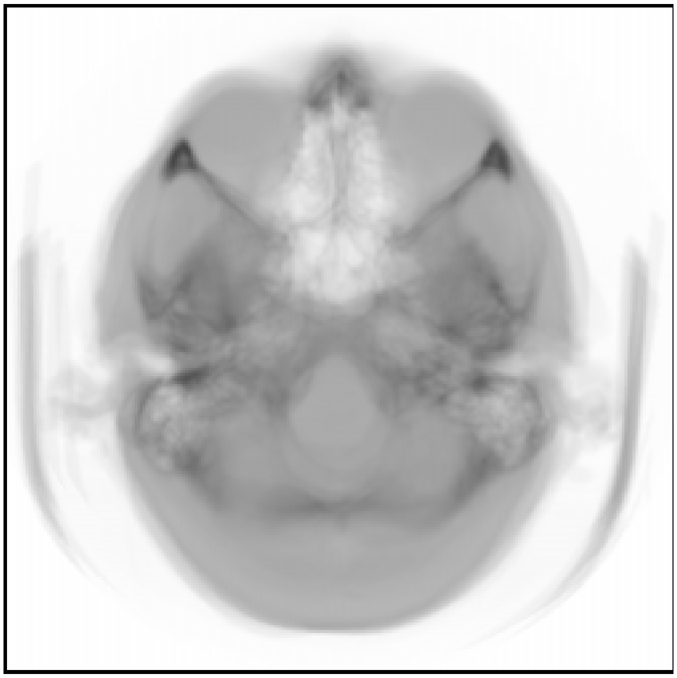


Figure 19: The “average” CT brain image

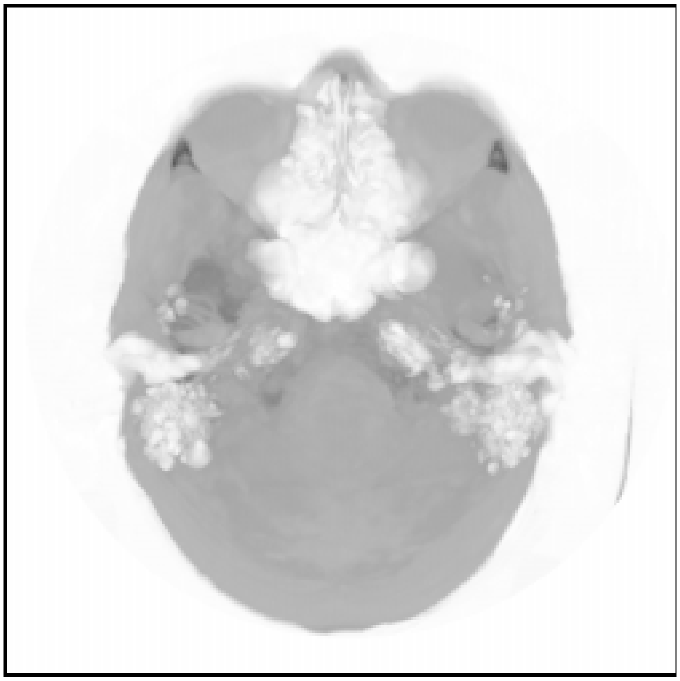


Figure 20: The “minimum” CT brain image

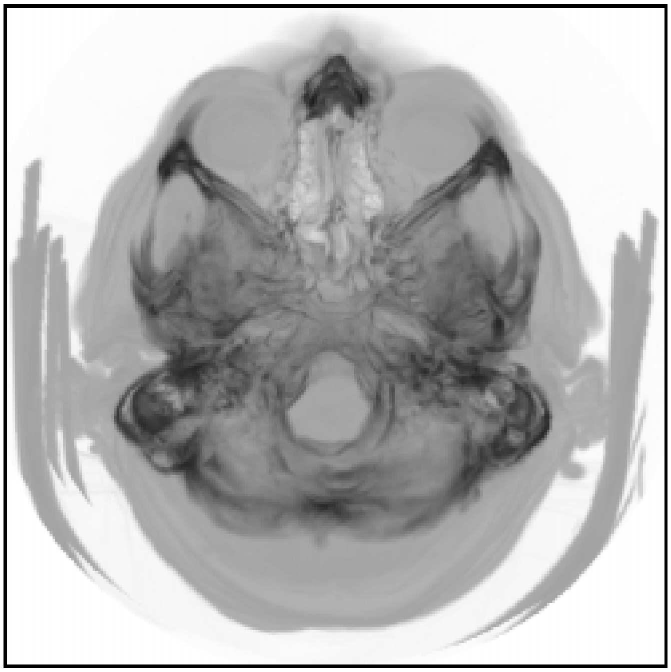


Figure 21: The “maximum” CT brain image

7.4.5.1 The "Min-Max Differential" (MMD) Method

This method was implemented as it is described in chapter 6. The results are presented in Table 2. The improvement in compression is defined as:

$$improvement = \frac{R_{SRC} - R}{R}$$

where R is the compression ratio achieved when using a regular compression method only, and R_{SRC} is the compression ratio achieved when combining *SRC* with that regular compression method. As the results show, the compression of all three compression methods was improved when they were combined with MMD.

Table 2: Experimental results from MMD method (CT images)

	Average size (Kbytes)	Lossless compression ratio	Comments
Original image	262	N/A	
Huffman compressed **MMD + Huffman**	190 128	1.379 : 1 2.047 : 1	 48 % improvement
Arithmetic compressed **MMD + Arithmetic**	154 120	1.701 : 1 2.183 : 1	 28 % improvement
Lempel-Ziv compressed **MMD + Lempel-Ziv**	107 95	2.449 : 1 2.758 : 1	 13 % improvement

7.4.5.2 The "Min-Max Predictive" (MMP) Method

Three different versions of the MMP method were used. In the first one, the prediction for the "level" of every pixel is set equal to the "level" of the previous pixel. This version has been named MMP1 (because it uses only 1 pixel for prediction), and the results are presented in Table 3.

Table 3: Experimental results from MMP1 method (CT images)

	Average size (Kbytes)	Lossless compression ratio	Comments
Original image	262	N/A	
Huffman compressed **MMP1 + Huffman**	190 90	1.379 : 1 2.911 : 1	 111 % improvement
Arithmetic compressed **MMP1 + Arithmetic**	154 88	1.701 : 1 2.977 : 1	 75 % improvement
Lempel-Ziv compressed **MMP1 + Lempel-Ziv**	107 86	2.449 : 1 3.047 : 1	 24 % improvement

In the second version of MMP (the MMP2), the prediction for the "level" of every pixel is calculated as the average value of the "levels" of its left and upper neighbor pixels. The results are presented in Table 4.

Table 4: Experimental results from MMP2 method (CT images)

	Average size (Kbytes)	**Lossless compression ratio**	**Comments**
Original image	262	N/A	
Huffman compressed **MMP2 + Huffman**	190 83	1.379 : 1 3.157 : 1	 129 % improvement
Arithmetic compressed **MMP2 + Arithmetic**	154 80	1.701 : 1 3.275 : 1	 93 % improvement
Lempel-Ziv compressed **MMP2 + Lempel-Ziv**	107 78	2.449 : 1 3.359 : 1	 37 % improvement

In the third version (the MMP3), the prediction is calculated by adding the "level" values of the upper and left neighbors, and then subtracting the "level" value of the upper-left neighbor. The results are presented in Table 5.

Table 5: Experimental results from MMP3 method (CT images)

	Average size (Kbytes)	**Lossless compression ratio**	**Comments**
Original image	262	N/A	
Huffman compressed **MMP3 + Huffman**	190 89	1.379 : 1 2.944 : 1	 113 % improvement
Arithmetic compressed **MMP3 + Arithmetic**	154 85	1.701 : 1 3.082 : 1	 81 % improvement
Lempel-Ziv compressed **MMP3 + Lempel-Ziv**	107 79	2.449 : 1 3.316 : 1	 35 % improvement

7.4.5.3 The "Centroid" Method

The main idea of the Centroid method is to use the "average" image as a predictor for the pixel values of the images to be compressed. As stated in section 6.3, this can be realized using various prediction models. A simplistic approach would be to subtract the values of the "average" image from each image to be compressed, and store only these differences. In section 6.3 a more sophisticated and improved approach was developed, based on a mathematical analysis. Both approaches were implemented and tested to demonstrate the difference in their performance. The first approach (taking the differences from the "average") is referred to as DiffAve, and the results are presented in Table 6. The second approach is simply referred to as the Centroid method, and the results are presented in Table 7. As it is shown, the Centroid method yields much better results than simply taking the differences from the "average" image.

Table 6: Experimental results from taking the differences from the average (CT images)

	Average size (Kbytes)	**Lossless compression ratio**	**Comments**
Original image	262	N/A	
Huffman compressed **DiffAve + Huffman**	190 162	1.379 : 1 1.617 : 1	 17 % improvement
Arithmetic compressed **DiffAve + Arithmetic**	154 146	1.701 : 1 1.795 : 1	 5 % improvement
Lempel-Ziv compressed **DiffAve + Lempel-Ziv**	107 107	2.449 : 1 2.449 : 1	 0 % improvement

Table 7: Experimental results from Centroid method (CT images)

	Average size (Kbytes)	**Lossless compression ratio**	**Comments**
Original image	262	N/A	
Huffman compressed **Centroid + Huffman**	190 100	1.379 : 1 2.620 : 1	 90 % improvement
Arithmetic compressed **Centroid + Arithmetic**	154 97	1.701 : 1 2.701 : 1	 59 % improvement
Lempel-Ziv compressed **Centroid + Lempel-Ziv**	107 91	2.449 : 1 2.879 : 1	 18 % improvement

7.4.5.4 Summary of Data

Table 8 presents a summary of the results for compressing sets of similar CT images. All SRC methods improved the compression ratios of the standard compression techniques that were tested with. The best improvement was obtained by the MMP2 method. Using this method, Huffman encoding improved by 129%, Arithmetic coding by 93%, and Lempel-Ziv by 37%.

Table 8: Summary of compression improvement using SRC methods (CT images)

	MMD	**MMP1**	**MMP2**	**MMP3**	**DiffAve**	**Centroid**
Huffman	48 %	111 %	129 %	113 %	17 %	90 %
Arithmetic	28 %	75 %	93 %	81 %	5 %	59 %
Lempel-Ziv	13 %	24 %	37 %	35 %	0 %	18 %

7.4.5.5 The Use of Robust Statistics

When regular statistics are used, outliers have a negative effect on minimum, maximum, and average values. This can adversely affect the performance of SRC methods. As it was described in section 6.4, robust statistics can be used to avoid the effect of outliers. When using robust statistics, the "average" image is replaced by the "median" image, whereas the "minimum" and "maximum" images are replaced by the "robust minimum" and "robust maximum" images (these are formed by excluding 10% of the extreme values). For the CT brain images used in the experiments, the "median" image is presented in Figure 22, the "robust minimum" in Figure 23, and the "robust maximum" in Figure 24.

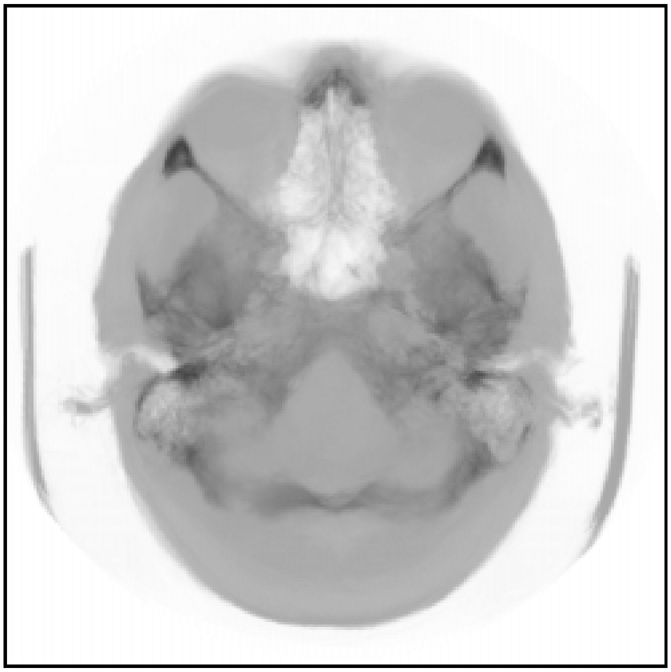


Figure 22: The "median" CT brain image

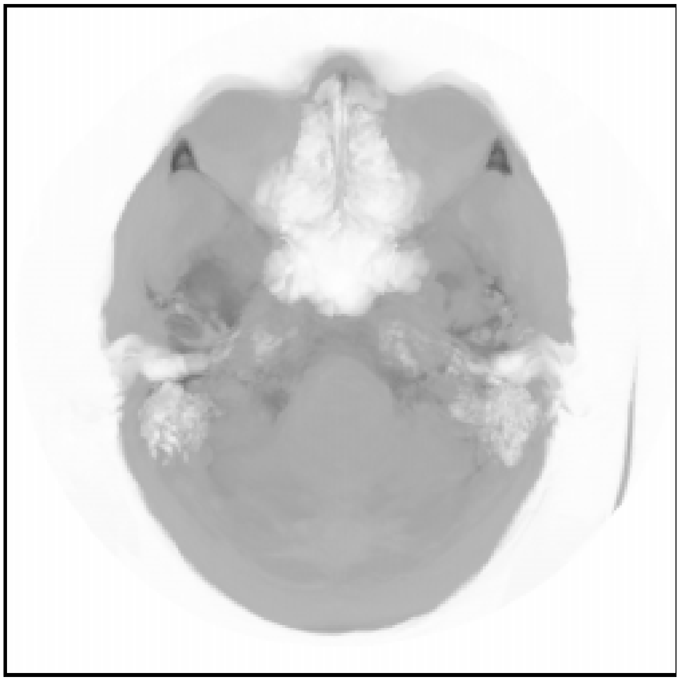


Figure 23: The “robust minimum” CT brain image

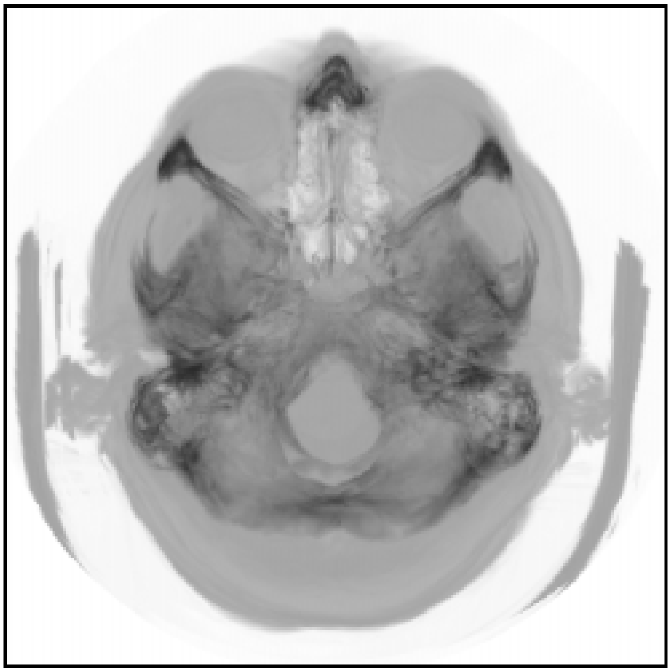


Figure 24: The “robust maximum” CT brain image

A summary of the results when robust statistics are employed is presented in Table 9. A comparison with the summarized results of Table 8 reveals that robust statistics do not improve compression over the regular statistics. This suggests that the test data did not contain enough outlier values to justify the additional implementation complexity associated with the use of robust statistics.

Table 9: Summary of compression improvement using SRC methods and robust statistics (CT images)

	MMD	**MMP1**	**MMP2**	**MMP3**	**DiffAve**	**Centroid**
Huffman	58 %	102 %	118 %	107 %	27 %	88 %
Arithmetic	33 %	67 %	81 %	73 %	12 %	57 %
Lempel-Ziv	4 %	16 %	27 %	26 %	0 %	11 %

7.4.6 Application of SRC Methods on MR Images

The set of MRI scans used in the experiments depict horizontal slices about 7-8 cm from the top of the head. In these slices the Cerebro-Spinal Fluid (CSF) in the middle looks loosely like an “X”. The “average”, “minimum”, and “maximum” images that were created for the experiments are presented in Figure 25, Figure 26, and Figure 27, respectively.

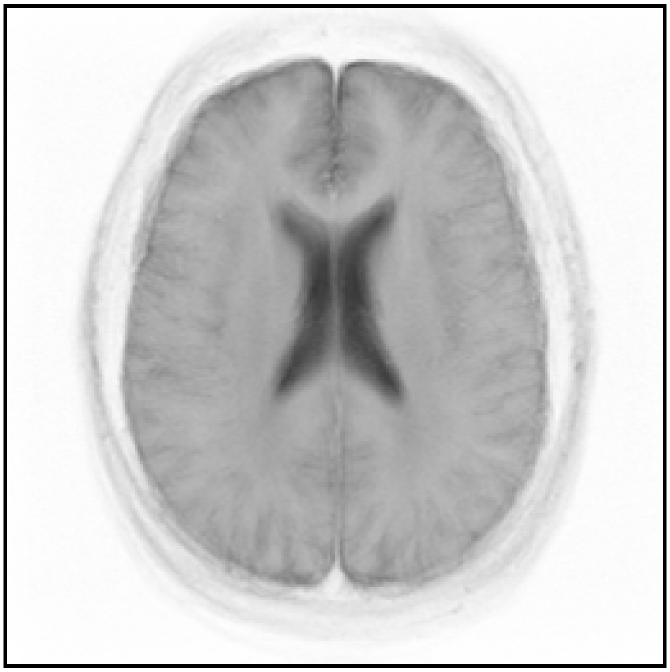


Figure 25: The “average” MR brain image

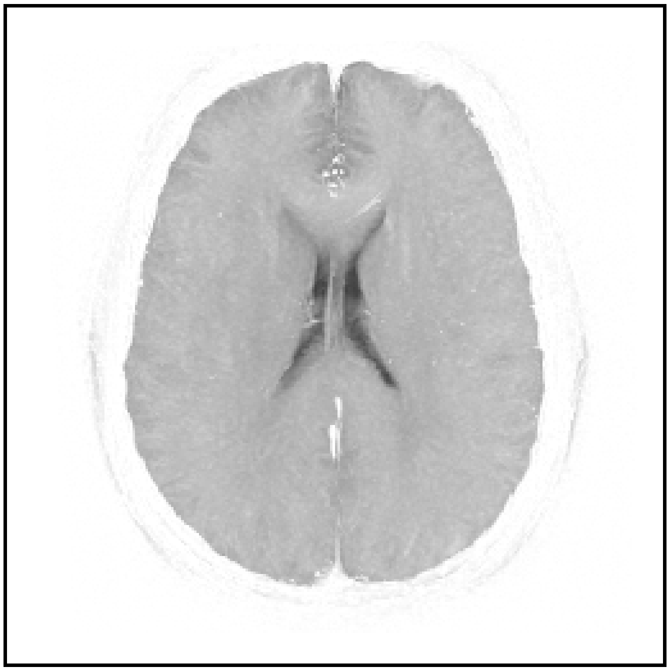


Figure 26: The “minimum” MR brain image

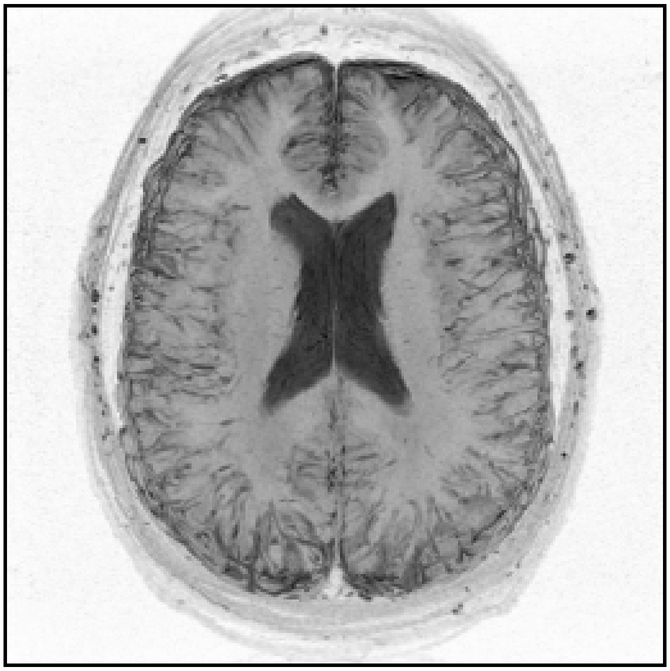


Figure 27: The “maximum” MR brain image

7.4.6.1 The “Min-Max Differential” (MMD) Method

The results from implementing the MMD method on MR images are shown in Table 10. The improvement in compression is smaller than the improvement achieved for implementing the MMD method on CT images (see Table 2). This can be attributed to the fact that MR images typically have lower signal-to-noise ratio than the CT images.

Table 10: Experimental results from MMD method (MR images)

	Average size (Kbytes)	**Lossless compression ratio**	**Comments**
Original image	66	N/A	
Huffman compressed **MMD + Huffman**	51 38	1.294 : 1 1.737 : 1	 34 % improvement
Arithmetic compressed **MMD + Arithmetic**	47 39	1.404 : 1 1.692 : 1	 21 % improvement
Lempel-Ziv compressed **MMD + Lempel-Ziv**	50 43	1.320 : 1 1.535 : 1	 16 % improvement

7.4.6.2 The “Min-Max Predictive” (MMP) Method

The three versions of the MMP method that were tested for the CT images were also used in the experiments with the MR images. Table 11 presents the results from implementing the MMP1 version, Table 12 the results from the MMP2, and Table 13 the results from the MMP3 version.

Table 11: Experimental results from MMP1 method (MR images)

	Average size (Kbytes)	**Lossless compression ratio**	**Comments**
Original image	66	N/A	
Huffman compressed **MMP1 + Huffman**	51 38	1.294 : 1 1.737 : 1	 34 % improvement
Arithmetic compressed **MMP1 + Arithmetic**	47 38	1.404 : 1 1.737 : 1	 24 % improvement
Lempel-Ziv compressed **MMP1 + Lempel-Ziv**	50 44	1.320 : 1 1.500 : 1	 14 % improvement

Table 12: Experimental results from MMP2 method (MR images)

	Average size (Kbytes)	Lossless compression ratio	Comments
Original image	66	N/A	
Huffman compressed	51	1.294 : 1	
MMP2 + Huffman	37	1.784 : 1	38 % improvement
Arithmetic compressed	47	1.404 : 1	
MMP2 + Arithmetic	37	1.784 : 1	27 % improvement
Lempel-Ziv compressed	50	1.320 : 1	
MMP2 + Lempel-Ziv	42	1.571 : 1	19 % improvement

Table 13: Experimental results from MMP3 method (MR images)

	Average size (Kbytes)	Lossless compression ratio	Comments
Original image	66	N/A	
Huffman compressed	51	1.294 : 1	
MMP3 + Huffman	37	1.784 : 1	38 % improvement
Arithmetic compressed	47	1.404 : 1	
MMP3 + Arithmetic	37	1.784 : 1	27 % improvement
Lempel-Ziv compressed	50	1.320 : 1	
MMP3 + Lempel-Ziv	42	1.571 : 1	19 % improvement

7.4.6.3 The “Centroid” Method

Similar to the experiments with the CT images, the DiffAve method (storing only differences from the “average” image) and the Centroid method were implemented. Table 14 presents the results from the DiffAve method and Table 15 the results from the Centroid method. Here too, the Centroid method clearly outperforms the DiffAve approach.

Table 14: Experimental results from taking the differences from average (MR images)

	Average size (Kbytes)	Lossless compression ratio	Comments
Original image	66	N/A	
Huffman compressed	51	1.294 : 1	
DiffAve + Huffman	45	1.467 : 1	13 % improvement
Arithmetic compressed	47	1.404 : 1	
DiffAve + Arithmetic	45	1.467 : 1	4 % improvement
Lempel-Ziv compressed	50	1.320 : 1	
DiffAve + Lempel-Ziv	49	1.347 : 1	2 % improvement

Table 15: Experimental results from Centroid method (MR images)

	Average size (Kbytes)	Lossless compression ratio	Comments
Original image	66	N/A	
Huffman compressed	51	1.294 : 1	
Centroid + Huffman	40	1.650 : 1	28 % improvement
Arithmetic compressed	47	1.404 : 1	
Centroid + Arithmetic	40	1.650 : 1	18 % improvement
Lempel-Ziv compressed	50	1.320 : 1	
Centroid + Lempel-Ziv	46	1.435 : 1	9 % improvement

7.4.6.4 Summary of Data

A summary of the results from implementing the SRC methods on the MR images is presented in Table 16. Comparing with the results of implementing the SRC method on the CT images (Table 8), it is clear that the noisy nature of MR images limits the compression improvements.

Table 16: Summary of compression improvement using SRC methods (MR images)

	MMD	MMP1	MMP2	MMP3	DiffAve	Centroid
Huffman	34 %	34 %	38 %	38 %	13 %	28 %
Arithmetic	21 %	24 %	27 %	27 %	4 %	18 %
Lempel-Ziv	16 %	14 %	19 %	19 %	2 %	9 %

7.4.6.5 The Use of Robust Statistics

The use of robust statistics was also tested in the case of MR image compression. The "median", "robust min", and "robust max" images are presented in Figure 28, Figure 29, and Figure 30, respectively. A summary of the results from using robust statistics with the SRC methods is presented in Table 17. Comparing Table 17 with Table 16 reveals that robust statistics do not improve the performance of the SRC methods for the MR images.

Table 17: Summary of compression improvement using SRC methods and robust statistics (MR images)

	MMD	MMP1	MMP2	MMP3	DiffAve	Centroid
Huffman	28 %	31 %	38 %	34 %	19 %	28 %
Arithmetic	15 %	18 %	24 %	24 %	7 %	15 %
Lempel-Ziv	6 %	9 %	16 %	14 %	4 %	9 %

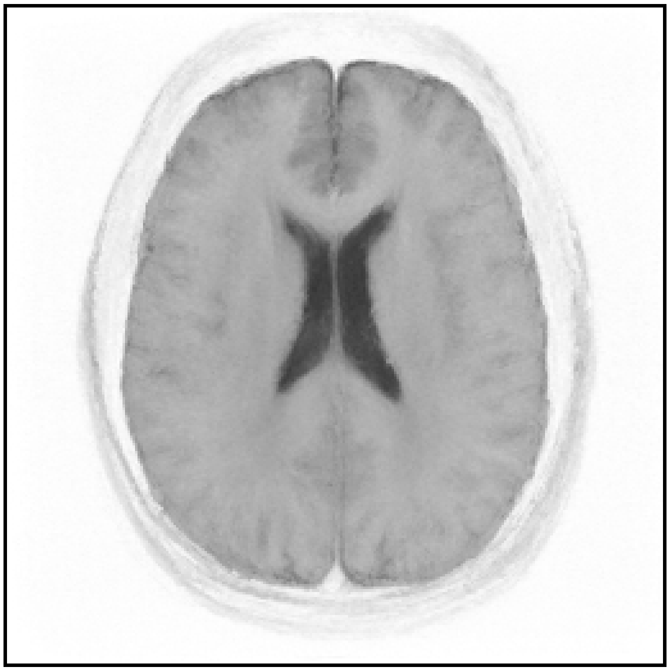


Figure 28: The “median” MR brain image

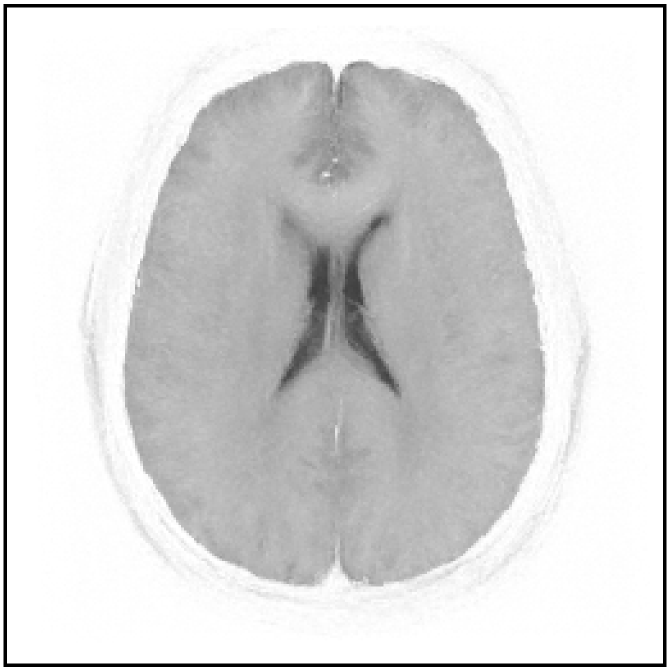


Figure 29: The “robust minimum” MR brain image

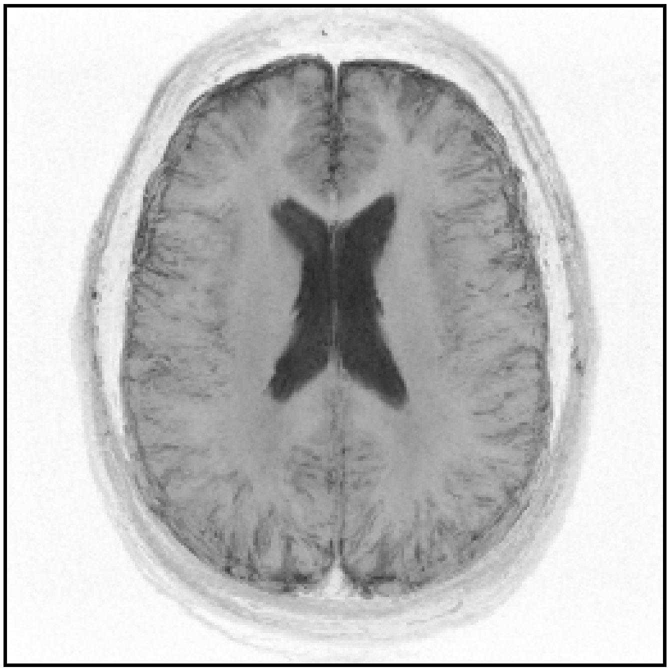


Figure 30: The “robust maximum” MR brain image

7.5 SUMMARY - CONCLUSIONS

As radiology becomes increasingly digital, the amount of digital imaging data produced in medical centers increases rapidly. This overtaxes the storage and communication capabilities of their computer systems and impedes the realization of an all-digital radiologic environment. Efficient compression methods can alleviate this problem by reducing the size of imaging data.

The standard procedures used in radiology result in images very similar to one another. For example, for every chest X-ray the position of the patient, the orientation of the imaging device, and the parameters used are standard. A collection of 1,000 chest X-rays is significantly more self-correlated, in the statistical sense, than a collection of random images. The same is true for any collection of medical images that are grouped together by modality and type of exam. This results in the existence of large amounts of set redundancy in medical image databases, that SRC methods can efficiently reduce to improve compression.

The implementation of SRC methods for medical image compression was demonstrated on a test database of 51 CT and 57 MRI brain scans. All these images were randomly selected from different patients and were registered in a standard position and size. Registration generally improves the effectiveness of the SRC methods because it increases the pixel-to-pixel similarities between images. However, registration is not necessary when the images are very similar to each other. In any case, omitting registration is expected to result in only a minor decrease in the effectiveness of the SRC methods.

Sets of similar images in the test database were formed by using a genetic algorithms procedure. In many cases, external information can be used to partition an image database into sets of similar images (for example, when textual data exist about the content of the images). In this research, no external information was used; instead, a general approach was sought and developed. The goal was to demonstrate how image clustering can be performed in the general case, when there are no external data about the content of the images.

Tests were performed by combining the SRC methods with three well-known compression techniques: Huffman, Arithmetic, and Lempel-Ziv compression. The results show that all SRC methods significantly improved the compression of these standard techniques. The best improvement (129%) resulted from combining the MMP method with Huffman compression. Figure 31 presents a bar chart with the percentage improvement that each SRC method produced when combined with each of the three standard compression techniques. This bar chart depicts the results from compressing CT images. Figure 32 presents a similar bar chart for MR images.

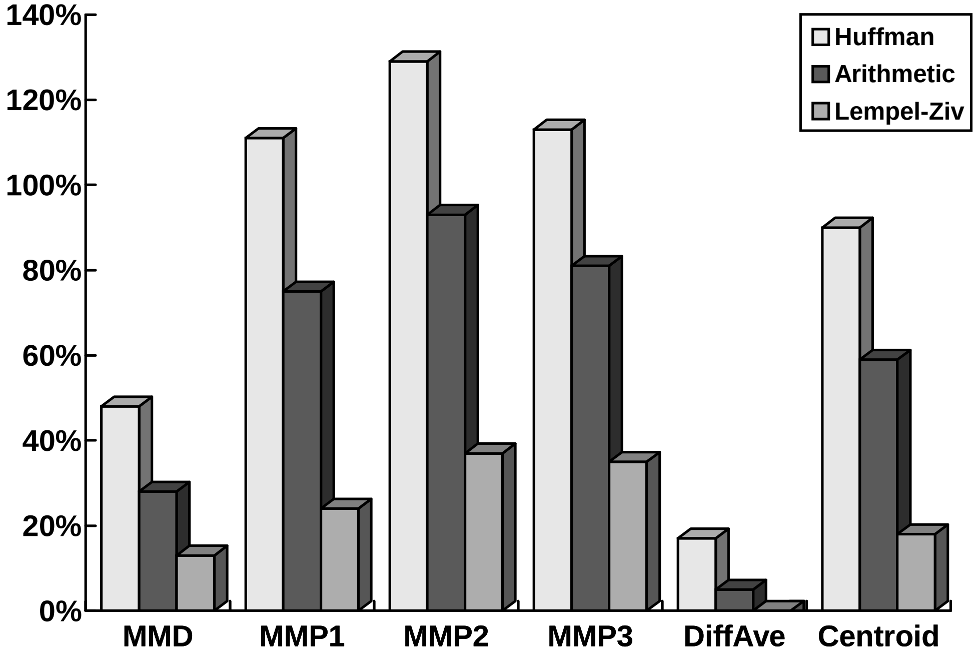


Figure 31: Improvement in compression by using SRC methods (CT images)

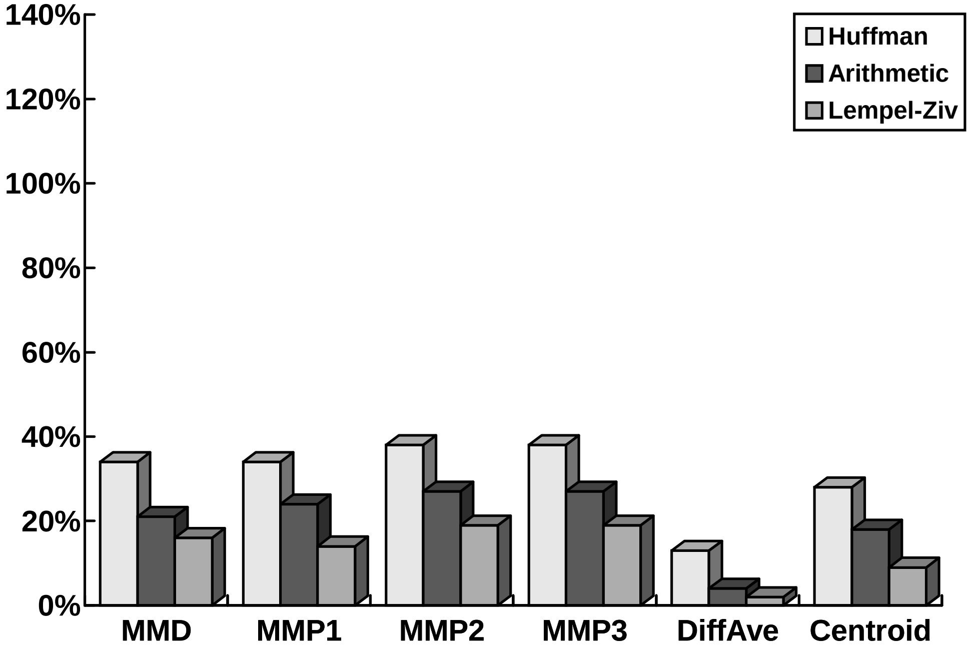


Figure 32: Improvement in compression by using SRC methods (MR images)

As shown in these two figures, the improvement in compression is smaller for the MR images. This can be attributed to the fact that MR images are very “noisy”, therefore although they have similar general features, the similarity at the pixel level is limited. As discussed in chapter 3, one of the characteristics of similar images is the similar pixel intensities in the same areas. MR images exhibit low signal-to-noise ratio, therefore there is large standard deviation in the values of every pixel position. For these reasons, SRC methods are not as effective on MR images as they are on the less noisy CT images. Nevertheless, SRC methods improved compression on MR images by as much as 38% for Huffman, 27% for Arithmetic, and 19% for the Lempel-Ziv method.

Finally, it is important to note the time performance of the SRC methods. All methods required less than 2 seconds of computation time on a Sun SPARC 20 workstation to process an image file of 262 Kbytes. Speed was one of the requirements for the development of compression methods based on the Enhanced Compression Model; all the SRC methods developed in this study satisfy this requirement.

CHAPTER 8

SUMMARY AND FUTURE DIRECTIONS

Image compression methods become increasingly important as new imaging applications are developed and the volume of imaging data grows rapidly. In many cases, large sets of similar images need to be compressed. However, the current image compression methods have been designed to compress *individual images* and do not take advantage of the potential for additional compression that exists in *sets of similar images*. This dissertation fills this research gap by setting the theoretical foundations and describing practical methods for effectively compressing sets of similar images.

Data compression is based on the existence of data redundancy. In individual images, three basic types of redundancy can be found: the interpixel, psychovisual, and symbol redundancy. All current compression methods are based on the same basic theoretical model that achieves compression by eliminating these three types of redundancy. However, sets of similar images contain an additional type of redundancy: the *set redundancy*. In this study the concept of set redundancy has been introduced and the *Enhanced Compression Model* has been developed to capitalize on set redundancy. This new model extends the current theoretical compression model by including an additional step for set redundancy extraction.

Based on the Enhanced Compression Model, sets of similar images are compressed by combining set redundancy extraction with regular compression techniques. Some of the requirements that practical set redundancy extraction must satisfy are the following:

- it must not be computationally expensive;
- it must enable compression and decompression of individual images from a set;
- it must be lossless.

Three methods that satisfy these requirements have been developed: the *Min-Max Differential (MMD)* method, the *Min-Max Predictive (MMP)* method, and the *Centroid* method. These methods and their variations are collectively referred to as *SRC (Set Redundancy*

Compression) methods. SRC methods increase the compressibility of the images in a lossless way; in addition, they are fast and easy to implement.

Medical imaging is one of the best application areas for the Enhanced Compression Model. Medical centers produce and store Terabytes of imaging data each year, and telemedicine applications transmit Gigabytes of imaging data every day. Medical images classified by modality and type of exam are very similar to one another, because of the standard procedures used in radiology. Therefore, medical image databases store large sets of similar images that contain great amounts of set redundancy. Experiments were performed to measure the improvement of compression that SRC methods can deliver in medical imaging. A test database of CT and MR brain scans from a random population of patients was used. The SRC methods were tested in combination with Huffman encoding, Arithmetic coding, and Lempel-Ziv compression. The results showed that SRC methods improved compression by as much as two-fold when compared to single image compression.

In practice, any database that stores sets of similar images can significantly benefit from the use of the Enhanced Compression Model. In addition to medical imaging, other potential application areas are the following:

- **Remote sensing**

  Satellite images that depict the same geographical area are usually remarkably similar to one another. If there are any differences, they are mainly due to meteorological variations (clouding coverage). However, the data for the morphology of the earth surface beneath the clouds remain more or less constant, and are common to every satellite photograph taken over that same area. These data represent set redundancy which is very large when images are taken with the same lighting conditions (i.e., same time of the day) and with similar weather conditions. SRC methods can effectively used to remove this set redundancy before the images are transmitted or stored.

- **Industrial image databases**

  Imaging technology in industry is often used for quality control that sometimes results in large databases of photographs depicting a single type of product or mechanical part. For example, in chemical plants pictures of pipelines are regularly taken and stored digitally in order to keep records of pipeline condition. Another example is manufacturing lines in which the products are monitored through robot vision applications and images of the products are stored for further computer processing. The result is image databases that store large numbers of very similar images.

- **Astronomical image databases**

  Astronomers closely monitor specific areas in the sky by repetitively taking pictures of them. These pictures record the relative movement of celestial objects over a period of time and are stored digitally in image databases. Such movement is relatively slow, and the change in these images is very small; consequently astronomical image databases represent another good application area for the Enhanced Compression Model.

- **Computer simulations in engineering**

  In many engineering fields, computer simulations are used to perform experiments and to test new designs. The output datasets of these simulations are often stored for further processing. In addition, the same simulations may run several times with small changes in parameters to test for different conditions. The output datasets that result from these simulations are often very similar to each other, especially when the changes in the parameters do not introduce large changes in the output. Even though these datasets are not images per se, the SRC methods can still be used to extract the set redundancy; then a general data compression method can be used to compress them.

The present research has focused on applying the Enhanced Compression Model in medical imaging. Future research may consider the application areas outlined above. Another open field for future investigation is the development of additional methods based on set redundancy. The three SRC methods developed in this study may be used as starting points for the development of other SRC techniques. Finally, the use of lossy image compression techniques may be explored as an alternative to the use of lossless techniques for the Enhanced Compression Model.

BIBLIOGRAPHY

[Algazi69] Algazi V.R., and Sakrison D.J., “On the optimality of the Karhunen-Loeve expansion”, *IEEE Trans. Information Theory*, pp. 319-321, 1969.

[Barnsley89] Barnsley M., *Fractals everywhere*, Academic Press, New York, 1989.

[Bassiouni85] Bassiouni M.A., “Data compression in scientific and statistical databases”, *IEEE Trans. on Software Engineering*, vol. 11, no. 10, pp. 1047-1058, Oct. 1985.

[Bramble89] Bramble J.M., “Comparison of information-preserving and information losing data-compression algorithms for CT images”, *Radiology*, vol. 170, pp. 453-455, Feb. 1989.

[Burt83] Burt P.J., and Adelson E.H., “The Laplacian pyramid as a compact image code”, *IEEE Trans. Commun.*, vol. COM-31, no. 4, pp. 532-540, April 1983.

[Chan89] Chan K.K., Lou S.L., Huang H.K., “Full-frame transform compression of CT and MR images”, *Radiology*, vol. 171, pp. 847-851, 1989.

[Chen94] Chen K., and Ramabadran T.V., “Near-Lossless Compression of Medical Images through Entropy-Coded DPCM”, *IEEE Trans. on Medical Imaging*, vol. 13, no. 3, Sep. 1994.

[Cheng88] Cheng Y., Iyengar S.S., and Kashyap R.L., “A new method of image compression using Irreducible Covers of Maximal Rectangles”, *IEEE Trans. on Software Engineering*, vol. 14, no. 5, pp. 651-658, May 1988.

[Chiariglione95] Chiariglione L., “The development of an Integrated Audiovisual Coding Standard: MPEG”, *Proc. IEEE*, vol. 83, no. 2, pp. 151-157, Feb. 1995.

[Chimiak92] Chimiak W.J., “The digital radiology environment”, *IEEE Journal on Selected Areas in Commun.*, vol. 10, no. 7, pp. 1133-1144, Sep. 1992.

[Chipman92] Chipman K., Holzworth P., Loop J., Ransom N., Spears D., and Thompson B., “Medical applications in a B-ISDN field trial”, *IEEE Journal on Selected Areas in Commun.*, vol. 10, no. 7, pp. 1173-1187, Sep. 1992.

[Cline93] Cline H.E., and Lorensen W.E., *Three Dimensional Medical Imaging: Methods and Applications*, General Electric Corporate Research and Development, Schenectady, NY, 1993.

[Cosman93] Cosman P.C., Tseng C., Gray R.M., Olshen R.A., Moses L.E., Davidson H.C., Bergin C.J., and Riskin E.A., "Tree-structured vector quantization of CT chest scans: image quality and diagnostic accuracy", *IEEE Trans. on Medical Imaging*, vol. 12, no. 4, pp. 727-739, Dec. 1993.

[Cox92] Cox J.R., Muka E., Blaine J.G., Moore S.M., and Jost G.R., "Considerations in Moving Electronic Radiography into Routine Use", *IEEE Journal on Selected Areas in Commun.*, vol. 10, no. 7, pp. 1108-1120, Sep. 1992.

[Das93] Das M., and Burgen S., "Lossless compression of medical images using two-dimensional multiplicative autoregressive models", *IEEE Trans. on Medical Imaging*, vol. 12, Dec. 1993.

[Dwyer92] Dwyer S.J., Templeton A.W., Anderson W.H., *et.al.*, "Teleradiology using Switched Dialup Networks", *IEEE Journal on Selected Areas in Commun.*, vol. 10, no. 7, pp. 1161-1172, Sep. 1992.

[Elnahas86] Elnahas S.E., Tzou K.H., Cox J.R., Hill R.L., Gilbert R.G., "Progressive coding and transmission of digital diagnostic pictures", *IEEE Trans. on Medical Imaging*, vol. MI-5, no. 2, pp. 73-83, June 1986.

[Gonzalez93] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, 1993.

[Hankamer79] Hankamer M., "A modified Huffman procedure with reduced memory requirement", *IEEE Trans. Commun.*, vol. COM-27, no. 6, pp. 930-932, 1979.

[Healy95] Healy D.M. Jr., and Weaver J.B., "Adapted Waveform Encoding for Magnetic Resonance Imaging", *IEEE Engineering in Medicine and Biology*, vol. 14, no. 5, pp. 621-638, Sep.-Oct. 1995.

[Hilton94] Hilton M.L., Jawerth B.D., and Sengupta A., "Compressing Still and Moving Images with Wavelets", *Multimedia Systems*, vol. 2, no. 3, 1994.

[Holland75] Holland J.H., *Adaptation in Natural and Artificial Systems*, Ann Arbor: The University of Michigan Press, 1975.

[Hotelling33] Hotelling H., "Analysis of a complex of statistical variables into principle components", *J. Educ. Psychol.*, vol. 24, pp. 417-441, 1933.

[Huang63] Huang J.J.Y., and Schultheiss P.M., "Block quantization of correlated Gaussian random variables", *IEEE Trans. on Communication Systems*, vol. CS-11, pp. 289-296, Sep. 1963.

[Huang92] Huang H.K., Wong W.K., Lou S.L., and Stewart B.K., "Architecture of a comprehensive radiologic imaging network", IEEE Journal on Selected Areas in Communications, vol. 10, no. 7, pp. 1188-1196, Sep. 1992.

[Huffman52] Huffman D.A., "A method for the construction of minimum redundancy codes", *Proc. IRE*, no. 40, pp. 1098-1101, Sep. 1952.

[Ishigaki90] Ishigaki T., Sakuma S., Ikeda M., Itoh Y., Suzuki M., and Iwai S., "Clinical evaluation of irreversible image compression: Analysis of chest imaging with computer radiography", *Radiology*, vol. 175, no. 3, pp. 739-743, June 1990.

[Jacquin92] Jacquin A., "Image coding based on a fractal theory of Iterated Contractive Image Transformations", *IEEE Trans. on Image Processing*, vol. 1, no. 1, p. 18, January 1992.

[Jain81a] Jain A.K., "Image data compression: a review", *Proc. IEEE*, vol. 69, pp. 349-389, 1981.

[Jain81b] Jain J.R., and Jain A.K., "Displacement measurement and its application in interframe image coding", *IEEE Trans. Commun.*, vol. COM-29, no. 12, pp. 1799-1808, Dec. 1981.

[Karadimitriou96] Karadimitriou K., Tyler J.M., "The Min-Max Differential Method for Large-Scale Storage and Compression of Medical Images", *Proc. of Annual Molecular Biology and Biotechnology Conference*, Baton Rouge, Feb. 1996.

[Karhunen47] Karhunen H., "Uber Lineare Methoden in der Wahrscheinlinchkeitsre-chnung", *Ann. Acad. Sci. Fennicae*, Ser. A137, 1947, (translated by I. Selin in "On Linear Methods in Probability Theory", T-131, 1960, the RAND Corp., Santa Monica, California).

[Karlsson88] Karlsson G., and Vetterli M., "Three-dimensional sub-band coding of video", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1100-1103, New York, 1988.

[Keizers92] Keizers A., Meyer-Ebrecht D., and Vosseburger F., "A fiber-optic line-switching network with a 140 Mb/s user data rate", *IEEE Journal on Selected Areas in Commun.*, vol. 10, no. 7, pp. 1197-1202, Sep. 1992.

[Knuth85] "Dynamic Huffman coding", *J. Algorithms*, no. 6, pp. 163-180, 1985.

[Kositpaiboon89] Kositpaiboon R., Tsingotjidis P., Barbosa L.O., and Georganas N.D., "Packetized Radiographic Image Transfers over Local Area Networks for Diagnosis and Conferencing", *IEEE Journal on Selected Areas in Commun.*, vol. 7, no. 5, pp. 842-856, June 1989.

[Kramer56] Kramer H.P., and Mathews M.V., "A linear encoding for transmitting a set of correlated signals", *IRE Trans. on Information Theory*, vol. IT-2, pp. 41-46, Sep. 1956.

[Kuduvalli92] Kuduvalli G.R., and Rangayyan R.M., "Performance analysis of reversible image compression techniques for high-resolution digital teleradiology", *IEEE Trans. on Medical Imaging*, vol. 11, pp. 430-445, Sep. 1992.

[Langdon84] Langdon G.G., "An introduction to arithmetic coding", *IBM J. Res. Develop.*, vol. 28, no. 2, pp. 135-149, March 1984.

[Legall91] LeGall D., "MPEG: a video compression standard for multimedia applications", *Commun. ACM*, vol. 34, no. 4, pp. 46-58, April 1991.

[Lelewer87] Lelewer D.A., and Hirshberg D.S., "Data compression", *ACM Comput. Surveys*, vol. 19, no. 3, pp. 261-295, 1987.

[Linde80] Linde Y., Buzo A., and Gray R.M., “An algorithm for vector quantizer design”, *IEEE Trans. on Commun.*, vol. COM-28, pp. 84-95, Jan. 1980.

[Liou91] Liou M., “Overview of the px64 kbit/s video coding standard”, *Commun. ACM*, vol. 34, no. 4, pp. 59-63, April 1991.

[Lo85] Lo S.C., and Huang H.K., “Radiologic image compression: Full-frame bit-allocation technique”, *Radiology*, vol. 155, pp. 811-817, 1985.

[Loeve48] Loeve M., “Fonctions Aleatoires de Second Ordre”, in P. Levy, *Processsus Stochastiques et Mouvement Brownien, Hermann*, Paris, 1948.

[Mallat89] Mallat S., “A theory for multiresolution signal decomposition: the wavelet representation”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674-693, July 1989.

[Mandelbrot77] Mantelbrot B.B., *The Fractal Geometry of Nature*, W.H. Freedman and Co., New York 1977.

[Mitchell88] Mitchell J.L., and Pennebaker W.B., “Software implementations of the Q-coder”, *IBM J. Res. Develop.*, vol. 32, no. 6, pp. 753-774, Nov. 1988.

[Moffat95] Moffat A., Neal R., Witten I.H., "Arithmetic Coding Revisited", *Proc. IEEE Data Compression Conference*, Snowbird, Utah, March 1995.

[Nasrabadi88] Nasrabadi N.M., and King R.A., “Image coding using vector quantization: a review”, *IEEE Trans. on Commun.*, vol. 36, no. 8, pp. 957-971, Aug. 1988.

[Neter89] Neter J., Wasserman W., and Kutner M.H., *Applied Linear Regression Models*, IRWIN, Burr Rigde, IL, 1989.

[Netravali80] Netravali A.N., and Limb J.O., “Picture coding: a review”, *Proc. IEEE*, vol. 68, no. 3, pp. 366-406, 1980.

[Okubo95] Ocubo S., “Reference Model Methodology - a tool for the collaborative creation of video coding standards”, *Proc. IEEE*, vol. 83, no. 2, Feb. 1995.

[Orozco92] Orozco-Barbosa L., Karmouch A., Georganas N.D., and Goldberg M., “A multimedia interhospital communications system for medical consultations”, *IEEE Journal on Selected Areas in Commun.*, vol. 10, no. 7, pp. 1145-1157, Sep. 1992.

[Ozcelik95] Ozcelik T., Brailean J.C., Katsaggelos A.K., “Image and Video Compression Algorithms based on Recovery Techniques using Mean Field Annealing”, *Proc. IEEE*, vol. 83, no. 2, pp. 304-316, Feb. 1995.

[Pennebaker88] Pennebaker W.B., Mitchell J.L., Langdon G.G., Jr., and Arps R.B., “An overview of the basic principles of the Q-coder adaptive binary arithmetic coder”, *IBM J. Res. Dev.*, vol. 32, no. 6, pp. 717-726, Nov. 1988.

[Pentland93] Pentland A., and Horowitz B., “A practical approach to fractal-based image compression”, in *Digital images and human vision*, Watson A.B., ed., MIT Press, pp. 53-59, 1993.

[Pirsch95] Pirsch P., Demassieux N., Gehrke W., “VLSI Architectures for Video Compression - A Survey”, *Proc. IEEE*, vol. 83, no. 2, pp. 220-246, Feb. 1995.

[Press92] Press W.H., Teukolsky S.A., Vetterling W.T., and Flannery B.P., *Numerical Recipes in Fortran: the art of scientific computing,* Cambridge University Press, New York, 1992.

[Rabbani91] Rabbani M, and Jones PW, *Digital Image Compression Techniques*, SPIE Press, Bellingham, Washington, USA, 1991.

[Ramabadran92] Ramabadran T.V., and Chen K., “The use of contextual information in the reversible compression of medical images”, *IEEE Trans. Medical Imaging*, vol. 11, no. 2, pp. 185-195, June 1992.

[Rhodes85] Rhodes M.L., Quinn J.F., and Silvester J., “Locally optimal run-length compression applied to CT image”, *IEEE Trans. on Medical Imaging.*, vol. MI-4, no. 2, pp. 84-90, June 1985.

[Rioul91] Rioul O., and Vetterli M., “Wavelets and Signal Processing”, *IEEE Signal Processing Magazine*, vol. 8, pp. 14-38, Oct. 1991.

[Riskin90] Riskin E.A., Lookabaugh T., Chou P.A., and Gray R.M., “Variable rate quantization for medical image compression”, *IEEE Trans. on Medical Imaging*, vol. 9, no. 3, pp. 290-298, Sep. 1990.

[Rissanen79] Rissanen J. and Langdon G.G., “Arithmetic coding”, *IBM J. Res. Develop.*, vol. 23, no. 2, pp. 149-162, March 1979.

[Roos88] Roos P., Viergever A., Van Dijke M.C.A., and Peters J.H., “Reversible intraframe compression of medical images”, *IEEE Trans on. Medical Imaging*, vol. 7, no. 4, pp. 328-336, Dec. 1988.

[Roos91] Roos P., and Viergever M.A., “Reversible interframe compression of medical images: A comparison of decorrelation methods”, *IEEE Trans. on Medical Imaging*, vol. 10, no. 4, pp. 538-547, Dec. 1991.

[Roos93] Roos P., and Viergever M.A., “Reversible 3-D decorrelation of medical images”, *IEEE Trans. on Medical Imaging*, vol. 12, no. 3, pp. 413-420, 1993.

[Saipetch95] Saipetch P., Ho B., Panwar R., Ma M., and Wei J., “Applying Wavelet Transforms with Arithmetic Coding to Radiological Image Compression”, *IEEE Engineering in Medicine and Biology*, vol. 14, no. 5, pp. 587-593, Sep.-Oct. 1995.

[Shannon48] Shannon C.E., “A mathematical theory of communication”, *Bell Syst. Tech. J.*, vol. 27, pp. 379-423, July 1948.

[Strang93] Strang G., "Wavelet transforms versus Fourier transforms", *Bulletin of the American Mathematical Society*, vol. 28, no. 2, pp. 288-305, April 1993.

[Strang94] Strang G., "Wavelets", *American Scientist*, vol. 82, pp. 250-255, May-June 1994.

[Tanaka82] Tanaka H., and Leon-Garcia A., "Efficient run-length encodings", *IEEE Trans. Inform. Theory*, IT-28(6), 880-890, 1982.

[Wallace91] Wallace G.K., "The JPEG Still Picture Compression Standard", *Commun. ACM*, vol. 34 , no. 4, pp. 30-44, April 1991.

[Welch84] Welch T.A., "A Technique for High Performance Data Compression", *IEEE Computer*, vol. 17, no. 6, pp. 8-19, June 1984.

[Wintz72] Wintz P.A., "Transform picture coding", *Proc. IEEE*, vol. 60, no. 7, pp. 809-820, July 1972.

[Witten87] Witten I.H., Neal R.M., Cleary J.G., "Arithmetic Coding for Data Compression", *Commun. ACM*, vol. 30, no. 6, pp. 520-540, June 1987.

[Wong92] Wong A.W., Taira R.K., and Huang H.K., "Implementation of a digital archive system for a radiology department", *Proc. SPIE Conf. on Medical Imaging VI: PACS Design and Evaluation* 1645, pp. 182-190, 1992.

[Wong95] Wong S., Zaremba L., Gooden D., and Huang H.K., "Radiologic Image Compression - A Review", *Proc. IEEE*, vol. 83, no. 2, pp. 194-218, Feb. 1995.

[Woods89] Woods J., and Naveen T., "Subband encoding of video sequences", *Proceedings of SPIE conference on Visual Communications and Image Processing*, vol. 1199, pp. 724-732, 1989.

[Yeo95] Yeo B.L., Liu B., "Volume rendering of DCT-based compressed 3D scalar data", *IEEE Trans. on Visualization and Computer Graphics*, vol. 1, no. 1, March 1995, pp. 29-43.

[Zhang92] Zhang Y.Q., and Zafar S., "Motion-compensated wavelet transform coding for color video compression", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 2, no. 3, pp. 285-296, Sep. 1992.

[Ziv77] Ziv J., and Lempel A., "A universal algorithm for sequential data compression", *IEEE Trans. Inf. Theory*, vol. IT-23, no. 3, pp. 337-343, May 1977.

[Ziv78] Ziv J., and Lempel A., "Compression of individual sequences via variable-rate coding", *IEEE Trans. Inf. Theory*, vol. 24, no. 5, pp. 530-536, Sep. 1978.

APPENDIX A

IMAGE ENTROPY AND COMPRESSION

Image entropy is defined in information theory [Shannon48] as the amount of information that an image contains. This "information content" establishes a limit to the maximum compression that can be achieved by symbol encoding the image. For example, if a given image has information content of 1,000 units and its current representation uses 20,000 units of information, then an optimum encoding scheme can achieve 1:20 lossless compression. This also implies that this image cannot be stored or represented with less than 1,000 units of information without losing some of its information content.

The entropy of an image can often be reduced by the use of an appropriate mapping or quantization. An appropriate mapping can sometimes transform a given image into another image with less entropy and with better compressibility than the original image. Quantization can achieve the same result, but with some loss of information.

Image entropy is calculated as follows [Gonzalez93]. Consider an information source S which generates a random sequence of symbols. There are $(n+1)$ possible symbols $\{a_0, a_1, ..., a_n\}$ that this source can generate. This set is called *source alphabet* and its elements are called *letters* or simply *symbols*. The source produces every symbol a_j with probability $\Pr(a_j)$. In addition:

$$\sum_{j=0}^{j=n} \Pr(a_j) = 1.0$$

The symbol generation from the source S can be considered to be a random event. In general, a random event E which occurs with probability $\Pr(E)$ contains $I(E)$ units of information, where:

$$I(E) = - \log \Pr(E).$$

The base of the logarithm in this equation determines the unit used to measure information. If the base is 2 then the unit is the bit. The average information from this source can be defined as

$$H = -\sum_{j=0}^{j=n} \mathrm{Pr}(a_j) \log \mathrm{Pr}(a_j)$$

and it corresponds to the entropy of the source S. H is a measure of the amount of uncertainty or information content associated with this source. The larger the value of the entropy H for a given source, the larger the amount of information it can deliver. The entropy is maximized when the source symbols have equal probability of occurrence, that is:

$$\mathrm{Pr}(a_0) = \mathrm{Pr}(a_1) = ... = \mathrm{Pr}(a_n) \quad <=> \quad H \text{ is maximum.}$$

Every 8-bit gray-level image can be assumed to be the output of an imaginary 8-bit gray-level source with alphabet {0, 1, 2, ..., 255}. The image entropy would be equal to the entropy of the source. This entropy can be calculated when the probabilities Pr(0), Pr(1), ..., Pr(255) are known. However, in general, these probabilities are unknown. The only known output from this imaginary source is its corresponding image. If this image is assumed to be a good statistical indicator of the source's behavior, then one can model the probabilities of the source symbols using the gray level histogram of the image. For example, consider an image with the following gray-level values:

5	30	30	24	20
5	24	24	24	20
5	5	5	24	20
5	5	20	20	20
30	30	30	30	30

For this image, the probabilities are:

Gray Level	Count	Probability
5	7	0.28
20	6	0.24
24	5	0.20
30	7	0.28
	25	1.00

Then, the entropy is

$$H = -(0.28 \log(0.28) + 0.24 \log(0.24) + 0.20 \log(0.20) + 0.28 \log(0.28))$$

that is

$$H = 1.9.$$

Note that the maximum possible entropy for a gray-level image with 256 gray levels is:

$$H_{\mathrm{max}} = 8.0.$$

The entropy can be used to estimate how many bits per pixel are required to represent the image. For the simple image in the example, $H = 1.9$ thus, this image can be coded with only 1.9 bits/pixel.

The computation of the entropy using this method is called *first-order entropy estimate*. Note that it does not take into account interpixel spatial relations (correlations) which in general decrease the information content (entropy) of an image. This results in an entropy estimate larger than the true value. Higher-order entropy estimates can be used for more accurate results. For a *second-order estimate*, instead of individual pixel frequencies the relative frequencies of 2-pixel blocks are used. Better approximations can be obtained with third, fourth, or even higher order estimates. However, the convergence of these estimates to the true entropy value is slow and computationally intensive. Therefore, despite its limitations, the first-order entropy estimate is usually preferred in practice because it provides a sufficiently good approximation to the entropy value with a minimum computational cost.

APPENDIX B

VIDEO COMPRESSION METHODS

B.1 Review of Methods

There are three general approaches in compressing a video sequence:

(a) Independent frame compression

Every frame is compressed individually using standard image compression methods. This is the simplest way, although it is not efficient because it does not take advantage of the temporal redundancy that exists between the successive frames.

(b) 3-D compression

In this approach the video sequence is considered to be a single 3-D image (where the third dimension is the time), and standard compression methods are used to compress it [Karlsson88]. In this way, the temporal redundancy is transformed into spatial redundancy which standard image compression can effectively use. However, there are two disadvantages. First, the 3-D image is usually too large to be handled efficiently by the current compression methods. Second, this method does not take into account motion information which can help significantly in compression.

(c) Predictive compression

Information from the preceding frames in the video sequence is used to create a prediction for the successive frames. Then only the errors between the prediction and the actual frames are stored. These error frames usually can be compressed better than the original frames, because they contain many zero values (where the prediction was successful). In addition, the small errors can be discarded, thereby further reducing the storage requirements.

The techniques based on predictive compression are the most successful ones for compressing video sequences, and therefore they have prevailed in today’s video compression methods. The simplest technique is to use frame difference coding: the differences between successive frames are calculated by simply subtracting the corresponding pixel values [Jain81a]. In this case the prediction for the next frame is simply its previous frame. This technique can be extended to use a linear combination of more than one of the previous frames to create a better prediction. However, more powerful techniques use also *motion compensation*[*] to improve the predictions [Netravali80]. These techniques use the previous frames to estimate motion parameters for the various objects in the current frame, and then according to these parameters (direction and velocity) they place the objects in the appropriate positions in the predicted frame. Jain [Jain81b] described a simple method to extract and use these motion parameters for video compression, and Woods [Woods89] presented various video compression algorithms and experimental results based on this idea.

The differences between the predicted and the actual frames are stored as error frames. Then standard image compression methods are used to reduce the spatial redundancy in these error frames; for example, DCT transform is used in MPEG video compression [Legall91]. More recently the wavelet transform has been used successfully [Zhang92, Hilton94].

Motion estimation is a difficult problem and a variety of approaches has been proposed to solve it. For example, Woods [Woods89] described motion estimation using subband encoding of video sequences. This hierarchical calculation of motion was found to work better than the direct calculation. Using this technique, a pyramid structure of images is constructed from every frame. A QMF filter is used to obtain the 11 (low-low) subband of the original frame, then the 11-11 subband is constructed, and so on. In other words, the image at a given level is a lower resolution version (subband 11) of the image just below it. A coarse estimation of motion is calculated using the lower subband, and thereafter this estimate is refined by using the higher subbands.

Other video compression methods perform image segmentation on every frame and then use the segmentation information to trace the displacement of moving objects across the images. However, these methods are quite complex and less effective for compression, because the boundaries of the segments as well as their translation need to be stored. A simpler method is to segment the images into fixed size blocks and then measure the translation of every one of these blocks. In this way only the displacement vector of each block needs to be stored [Jain81b].

Recently, Ozcelik, Brailean and Katsaggelos [Ozcelik95] proposed a new paradigm for designing encoders-decoders for video compression. According to this paradigm, the decoder should not simply undo the operations performed by the encoder but should solve an estimation problem to reconstruct the video sequence with the best possible quality, based on the compressed data. In other words, instead of the decoders performing passively the inverse operations done by the encoders, they actively try to reduce the artifacts due to the compression process and estimate the original video sequence, based on its (lossy) compressed version.

[*] *motion compensation*: utilization of the knowledge of motion or displacement of objects to predict their exact position in successive frames

B.2 The H.261 Video Compression Standard

The H.261 standard is the result of the collaboration of experts from 12 countries. The committee was formed on December 1984 and finished with the standardization process in November 1989. This standard was developed mainly for video phone and videoconferencing applications.

DESCRIPTION

Every input frame is divided into 8x8 blocks of pixels, and motion compensated predictive compression based on block motion between successive frames is used. The error frames are encoded using Discrete Cosine Transform (DCT) and then are quantized by a linear quantizer with adjustable step size (the step size depends on how full the transmitting buffer is; if the buffer is nearly full, the quantization increases, otherwise it decreases). For more details see [Liou91] [Okubo95].

B.3 The MPEG Video Compression Standard

This standard was developed by the Moving Picture Experts Group. This committee was created in 1988 by Leonardo Chiariglione. The MPEG development followed three phases:

(a) *Requirements Phase*: a proposal package description and a test methodology were developed.

(b) *Competition Phase*: companies and institutions submitted proposals for video compression methods and formats.

(c) *Convergence Phase*: collaborative process where ideas and techniques identified as promising at the end of the Competition Phase were integrated into one solution.

Two variations of the same standard emerged from the MPEG committee: the MPEG-1 and the MPEG-2. The main difference between MPEG-1 and MPEG-2 is that MPEG-1 has been optimized for noninterlaced format while MPEG-2 is a generic standard that can also handle the interlaced format [Pirsch95].

DESCRIPTION

Frames are classified as either Intraframes (I), Predicted (P), or Bidirectionally predicted (B) frames. The I frames are compressed individually by using DCT. Both P and B frames are compressed using a block-based motion compensated predictive technique. The difference between P and B frames is that for the prediction of P frames only previous frames are used, whereas for predicting the B frames both previous and successive frames are used. The I frames follow a standard distribution, for example 1 in every 10 frames. However the decision whether a frame will be encoded as P or B is made after testing it in both ways, and choosing the one with better compression.

The motion compensated prediction technique that MPEG uses is the following. Every input frame is divided into 16x16 blocks (called Macroblocks) and a motion vector is calculated for every block. These motion vectors and the proper input frames are used to create a prediction frame P or B. The errors between the predicted frames and the actual frames are compressed using the following sequence of operations: DCT - Quantization - Zigzag scan - Run length - Huffman encoding. In addition the motion vectors are encoded using variable length codes and stored together with the error frames and some additional overhead information. For more details see [Legall91, Chiariglione95].

Table 18: Summary of video compression standards

	H.261	**MPEG-1**	**MPEG-2**
Committee	CCITT	ISO	ISO
Applications	video phone, video conference	CD-ROM, computer applications	video distribution
Image format	QCIF, CIF (10 - 30 Hz)	SIF (25 Hz, 30 Hz)	CCIR
Bit rate	p x 64 Kb/s (1 ≤ p ≤ 30)	1.2 Mb/s	4 - 9 Mb/s

APPENDIX C

MEDICAL IMAGING MODALITIES

C.1 Computed Tomography (CT)

A Computed Tomography (CT) scanner consists of an X-ray source that rotates around the patient and has a linear X-ray detector array on the opposite side. This provides a single slice; a series of slices can be acquired by moving the patient. A new variation of the traditional CT is the helical CT. In this variation the device moves continuously around the patient in a helical path, producing as many as 100 image slices in 20 seconds [Cline93].

CT is the modality of choice for imaging bones and calcifications. The reason is that contrast in CT is a function of the electron density of tissue. Soft tissues have very similar electron densities; bone is the only tissue in the body with significantly different electron density. Therefore, normally CT has very limited soft tissue contrast. However, if contrast material is injected into the blood, soft tissues can also be contrasted. The main advantage of CT over other medical imaging methods is its high spatial resolution and accuracy of the data. A typical CT slice has 512x512 pixels, 0.5 mm resolution, and 1.5 mm thickness [Cline93].

C.2 Magnetic Resonance Imaging (MRI)

Magnetic Resonance Imaging (MRI) is a modality that best depicts soft tissue. The MR scanner consists of a magnet with gradient control coils, an RF section to excite nuclear magnetic spins, and a receiver. The contrast mechanism depends on the relaxation parameters of the water in tissue; different soft tissues have different spin relaxation times. For example, the white matter in the brain has 140% higher MR signal than gray matter (in comparison, the difference between white and gray matter in CT imaging is only about 12%) [Healy95]. Another advantage of MRI is the ability to obtain views at arbitrary positions and orientations. The disadvantages are the limited spatial resolution (typically MR images are only 256x256 pixels) and the long time required for an MR exam. Also the cost of an MRI exam is higher than the cost of a CT exam.

C.3 Emission Tomography (SPECT, PET)

Emission Tomography includes two different techniques: the Single Photon Emission Computed Tomography (SPECT), and the Positron Emission Tomography (PET). Both techniques give information about local tissue metabolism, which corresponds to functional activity. In general, a substance with radioactive molecules is injected into the blood stream of a patient; then the emitted X-rays are measured by a detector surrounding the object. The difference between these two techniques is that in SPECT only one photon is emitted from each molecule, whereas in PET a positron is emitted, which gives rise to two photons moving in opposite directions.

The advantage of these techniques is that radiopharmaceuticals can be used to selectively access specific parts of the human body. Essentially, both techniques visualize the concentration of the emitting material in different parts of the body. Since this concentration depends on the blood flow, areas with most activity (higher blood flow) are highlighted. This is very useful to visualize, in vivo, parts of the brain with the highest activity. Many psychological experiments use PET to identify which parts of the brain are activated when the subject performs certain tasks.

The resolution of SPECT is typically 6 mm, and resolution of PET is about 1 cm. The typical size of Positron Emission images is 128x128 pixels.

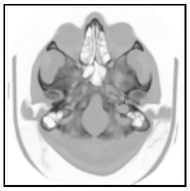
C.4 Ultrasound

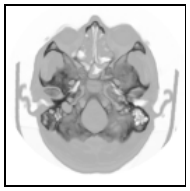
Ultrasound is an inexpensive and easy to use medical imaging technique. It uses a transducer that generates pulses of ultrasound and then detects the echo in the body. This technique can provide images in real time and is useful to visualize motion. The disadvantage is that ultrasound cannot depict structures deep in the body, and it cannot penetrate bone or air pockets surrounding an object of interest. This technique is useful mostly for visualization of the abdomen area and the heart.

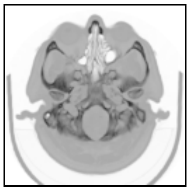
APPENDIX D

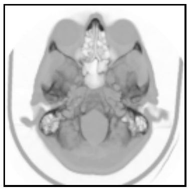
MEDICAL IMAGES USED IN THE EXPERIMENTS

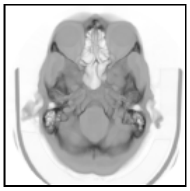
D.1 CT images

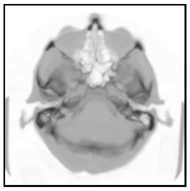

CT01

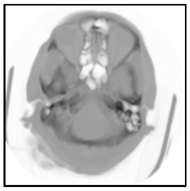

CT02

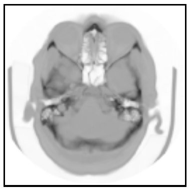

CT03

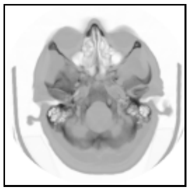

CT04

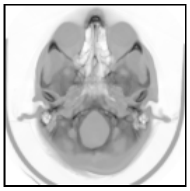

CT05

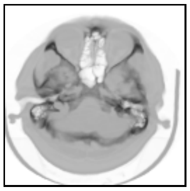

CT06

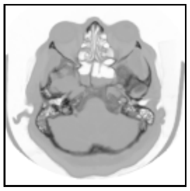

CT07

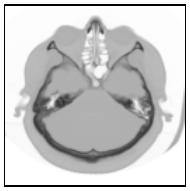

CT08

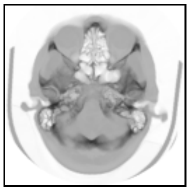

CT09

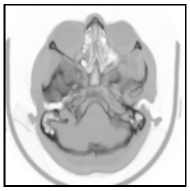

CT10

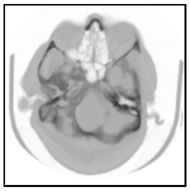

CT11

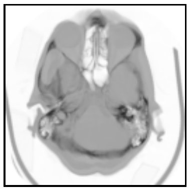

CT12

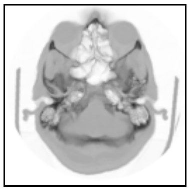

CT14

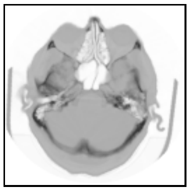

CT15

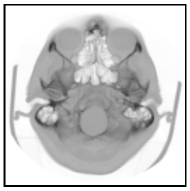

CT16

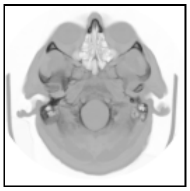

CT17

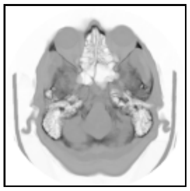

CT18

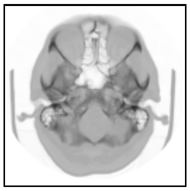

CT19

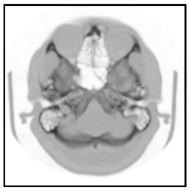

CT20

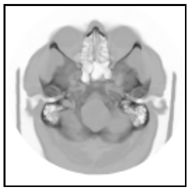

CT21

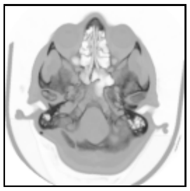

CT22

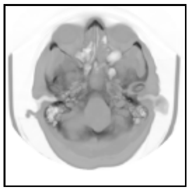

CT23

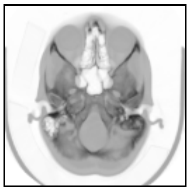

CT24

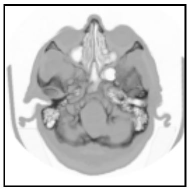

CT25

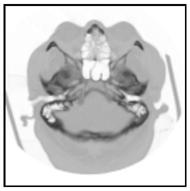

CT26

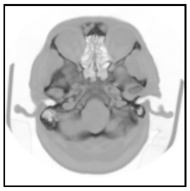

CT27

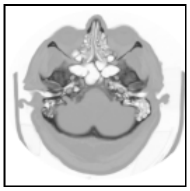

CT28

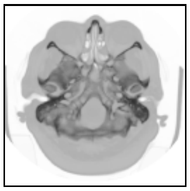

CT29

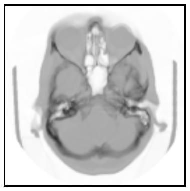

CT30

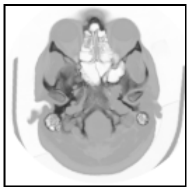

CT31

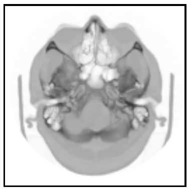

CT32

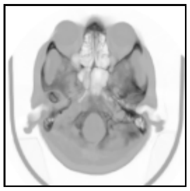

CT33

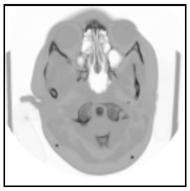

CT34

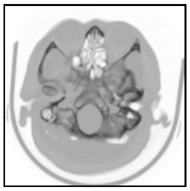

CT35

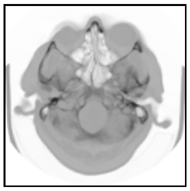

CT36

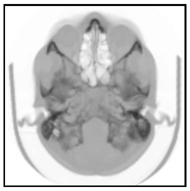

CT37

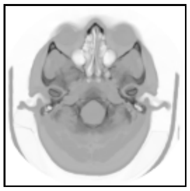

CT38

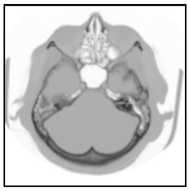

CT39

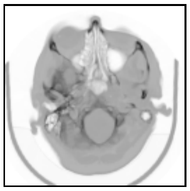

CT40

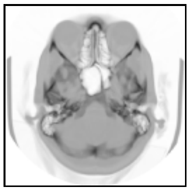

CT41

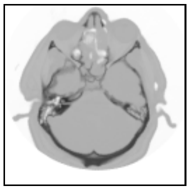

CT42

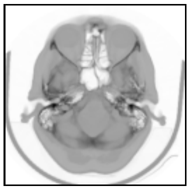

CT43

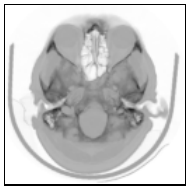

CT44

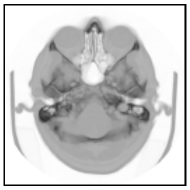

CT45

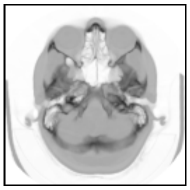

CT46


CT47

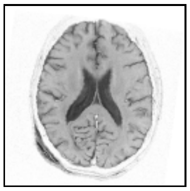

CT48

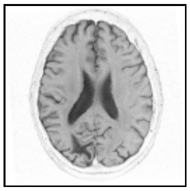

CT49

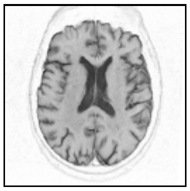

CT50

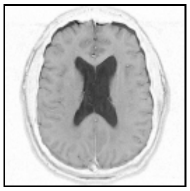

CT51

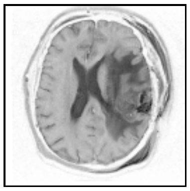
D.2 MR images

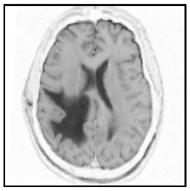

MR01

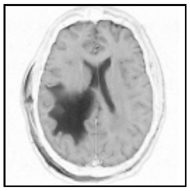

MR02

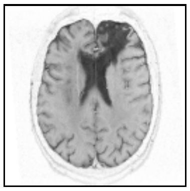

MR03

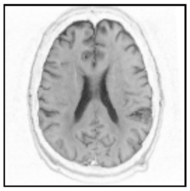

MR04

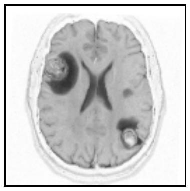

MR05

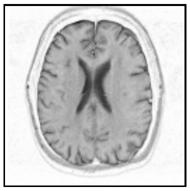

MR06

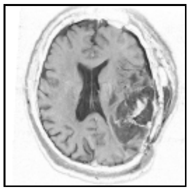

MR07

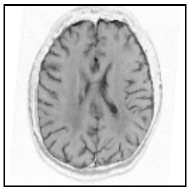

MR08

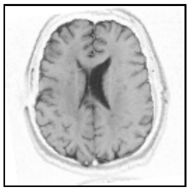

MR09

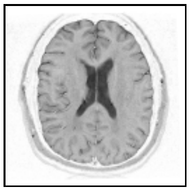

MR10

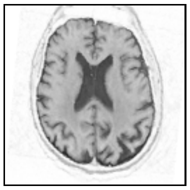

MR11

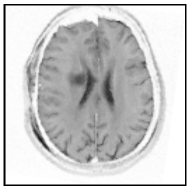

MR12

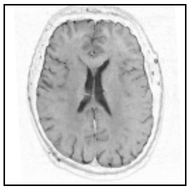

MR13

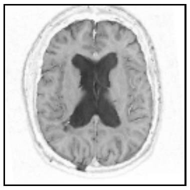

MR14

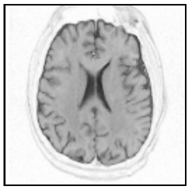

MR15

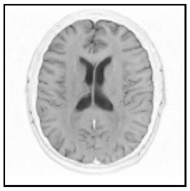

MR16

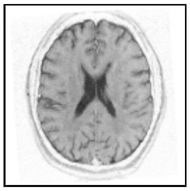

MR17

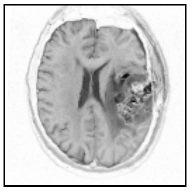

MR18

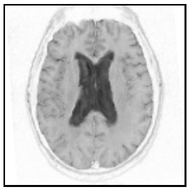

MR19

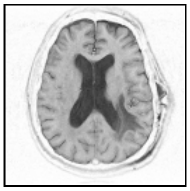

MR20

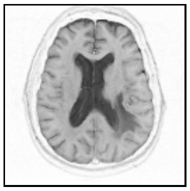

MR21

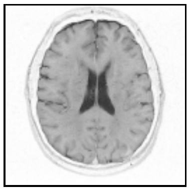

MR22

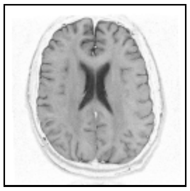

MR23

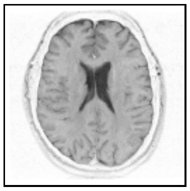

MR24

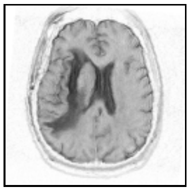

MR25

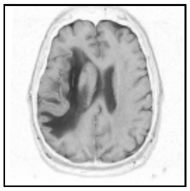

MR26

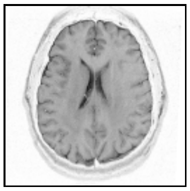

MR27

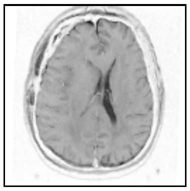

MR28

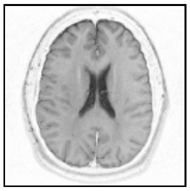

MR29

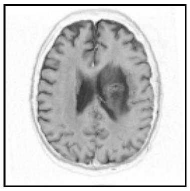

MR30

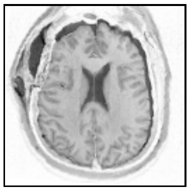

MR31

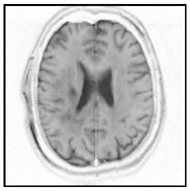

MR32

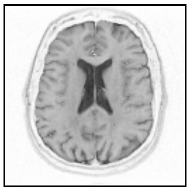

MR33

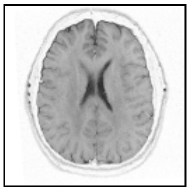

MR34

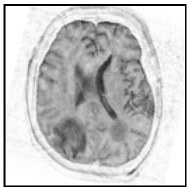

MR35

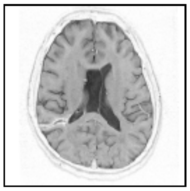

MR36

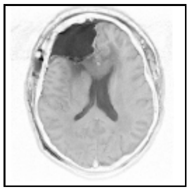

MR37

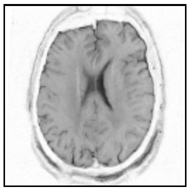

MR38

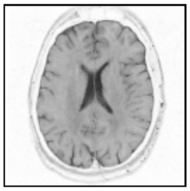

MR39

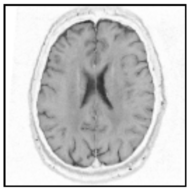

MR40

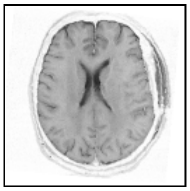

MR41

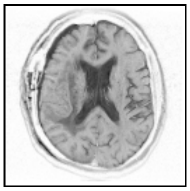

MR42

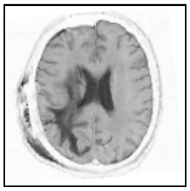

MR43

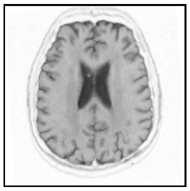

MR44

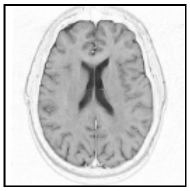

MR45

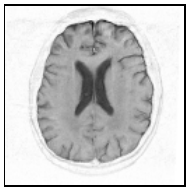

MR46

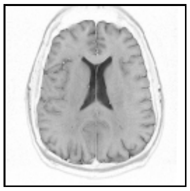

MR47

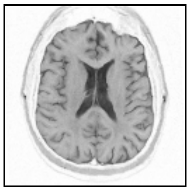

MR48

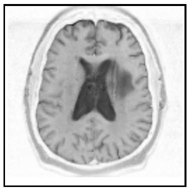

MR49

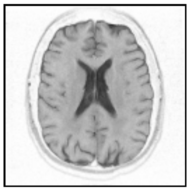

MR50

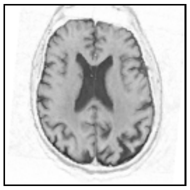

MR51

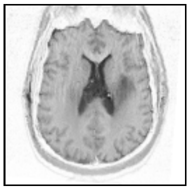

MR52


MR53


MR54


MR55


MR56


MR57

VITA

Kosmas Karadimitriou was born in Thessaloniki, Greece, in 1965. He received his Bachelor’s degree in Physics from Aristotle University of Thessaloniki, Greece, in 1988 and in 1989 he enlisted in the Greek Air Force for his mandatory military service. He served for 2 years as a second lieutenant and after his discharge he left Greece to pursue graduate studies in Computer Science at Louisiana State University, Baton Rouge, Louisiana. He received his Master’s degree from LSU in 1994. His research interests include Image Processing, Data Compression, Medical Imaging, Parallel Computing, and Biomedical applications of Computer Science.

IMPORTANT NOTE

This document is not identical to my dissertation, even though the content is exactly the same. There are changes in the format that make it easier to read and print it. It differs from my dissertation in two ways:

a) the format is more compact than in my original dissertation. Specifically, the font is smaller (11 pts instead of 12 pts), the line spacing is smaller (1.3 lines instead of double-space), and the page margins have also been decreased. As a result, this document is only 94 pages long, versus 132 pages in the original dissertation. Also, a header has been added in every page with the title of the chapter the page belongs to.

b) a more important change has been made to the images: all MR and CT images in this document have their greyscales inverted (which means, they are the negatives of the originals). There are a couple of reasons for making this change. First, because this definitely improves the visual clarity for this particular type of images: their histograms are heavily skewed towards smaller pixel values, which makes the originals appear on paper as a black mess. And second, because the images in my original dissertation were not particularly printer-friendly; every time I printed the dissertation I needed a new toner for the printer. So I decided to spare you from the trouble, and just invert the image colors - believe me, it's worth it!

- Kosmas Karadimitriou, 1/25/99