Evaluate Multiple Description Coding as an image processing method for transferring information in error-prone networks with low transmission rate, related to quality, bit rate and file size

by

Simen Torp Johnsen
Morten Standeren

Master's Thesis in
Information and Communication Technology

Agder University College
Faculty of Engineering and Science

Grimstad

May 2005

Abstract

Over the past several years, image transmissions and other multimedia services have become increasingly popular. The problem is that many of these services are performed in so-called unreliable transmission systems, for example wireless or mobile systems, where there is a greater probability of packet loss, bit errors and/or interference when compared to wired transmission systems.

In today's transmission systems, retransmission is a commonly used technique for the correction of lost, or erroneous, data packets. However, this technique has several shortcomings since the retransmissions often lead to a considerable amount of extra network traffic, which in turn results in a reduced availability of bandwidth for each user, and, at worst, may result in network congestion. The time spent on data retransmissions may also be a problem in several applications.

This thesis evaluates MD[1] coding as an alternative solution to the above-mentioned problems. MD coding is a technique to achieve robust communication over unreliable channels such as a lossy packet network. A source is split into two or more equally important descriptions in such a way that various reconstruction qualities are obtained from different subsets of the descriptions. In MD coding, statistical redundancy is added to a data signal in such a way that data packets, which are lost or exposed to errors during transmission, can be estimated from all, or some of, the successfully received data packets. This evaluation is accomplished by development and implementation of several SD[2] and MD coding algorithms, where the main objective is to prove that the MD coding algorithms may considerably reduce the need for retransmission in unreliable transmission systems. Since the available bandwidth in such systems is often limited, compression of image data is also an important part of this thesis.

The developed coding algorithms and compression procedures are implemented through a test application, which simulates image network transmissions with user-defined parameters. Finally, the efficiency of each developed MD coding algorithm is proved by comparing them with the SD coding systems, the so-called baseline systems.

Through this thesis project, it has been discovered, by means of qualitative and quantitative tests, that the MD coding systems outperform the SD coding systems when packet loss occur during image transmission, while matching the SD coding systems when all the image data are successfully transmitted. This is done by adding a certain amount of redundancy to the MD data signals.

[1] Multiple Description

[2] Single Description

Preface

This thesis is the last part of the master degree in Information and Communication Technology at Agder University College, faculty of engineering and science. The work was carried out between January 2005 and May 2005. The thesis was given by Agder University College and no external company was involved.

We would like to thank our teaching supervisor, Ola Torkild Aas at Agder University College, for valuable counselling and inspiration during the process of this thesis. We would also like to thank our director of studies, Stein Bergsmark, for useful tips and advices during this thesis project.

Grimstad, May 2005

Simen Torp Johnsen and Morten Standeren

Table of Contents

Abstract ..... *II*

Preface ..... *III*

Table of Contents ..... *IV*

List of Figures ..... *VI*

List of Equations ..... *VII*

List of Tables ..... *VII*

1 Introduction *1*

1.1 Thesis Definition **1**

1.2 Background **1**

1.3 Related Work **3**

1.4 Work Description **3**

1.5 Report outline **5**

2 Research Methodology *6*

2.1 Introduction **6**

2.2 Qualitative and Quantitative Research **6**

2.3 Summary **7**

3 The System *8*

3.1 Introduction **8**

3.2 System Overview **8**

4 Theory *10*

4.1 Introduction **10**

4.2 Compression **10**

4.2.1 Introduction 10
4.2.2 Color Space Transformation 11
4.2.3 Downsampling of Color Components 12
4.2.4 Discrete Cosine Transform 15
4.2.5 Quantization 17
4.2.6 Zigzag Coding 20
4.2.7 Run Length Coding 22
4.2.8 Entropy Coding 23
4.2.9 Wavelet Transform versus DCT 24

4.3 Coding Systems **26**

4.3.1 Introduction 26
4.3.2 Baseline Systems 27
4.3.3 Multiple Description Coding Systems 31
4.3.4 Comparison System 40

4.4 Quality Measurement **41**

4.4.1 Introduction 41
4.4.2 Mean Squared Error and Peak Signal to Noise Ratio 42
4.4.3 A Universal Image Quality Index 43

4.4.4 Mean Structural Similarity Index 46

5 Implementation *48*

5.1 Introduction **48**

5.2 The C# language and the .NET framework **48**

5.3 Test application **50**

5.3.1 Application Design 51

5.3.2 Graphical User Interface and Functionality 52

6 Results *54*

6.1 Introduction **54**

6.2 Test Procedure **54**

6.3 PSNR versus MSSIM **56**

6.4 Results of image processing with different coding techniques **57**

6.4.1 Baseline Systems 57

6.4.2 MD Coding Systems 62

6.5 Results of image processing at different compression rates **68**

6.6 Summary **69**

6.6.1 Theoretical Summary 69

6.6.2 Visual Summary 71

7 Discussion *74*

7.1 Introduction **74**

7.2 Results of image processing with different coding techniques **75**

7.2.1 Baseline Systems 75

7.2.2 MD Coding Systems 77

7.3 Results of image processing at different compression rates **78**

7.4 Future Work **79**

7.4.1 Introduction 79

7.4.2 General 79

7.4.3 Compression....80

7.4.4 MD Coding....80

7.4.5 Quality Measurement 80

7.4.6 Implementation....80

7.4.7 Future Work Summary 81

8 Conclusion *82*

Abbreviations *83*

Bibliography *84*

Appendix A: Huffman Tables *90*

Appendix B: Coding and Decoding Sequence *95*

Appendix C: Summary of System Performances *97*

Appendix D: One versus Both Descriptions Comparison *98*

Appendix E: C# source code and test images *99*

List of Figures

FIGURE 1-1: WIRED VS. WIRELESS TRANSMISSION SYSTEMS....2
FIGURE 4-1: DOWNSAMPLING OF CHROMINANCE COMPONENTS....12
FIGURE 4-2: UPSAMPLING OF CHROMINANCE COMPONENTS....13
FIGURE 4-3: WEIGHTED UPSAMPLING OF CHROMINANCE COMPONENTS....13
FIGURE 4-4: INPUT (LEFT) AND OUTPUT (RIGHT) DATA OF THE ZIGZAG PROCEDURE....21
FIGURE 4-5: RUN LENGTH CODING....22
FIGURE 4-6: BASELINE SYSTEM OVERVIEW....27
FIGURE 4-7: ILLUSTRATION OF THE SDTC SPLITTING....28
FIGURE 4-8: ILLUSTRATION OF THE SDDDC SPLITTING....29
FIGURE 4-9: MDPCT OVERVIEW....31
FIGURE 4-10: ILLUSTRATION OF THE MDPCT TRANSFORMATION....31
FIGURE 4-11: MDSQ OVERVIEW....35
FIGURE 4-12: MDSQ ENCODER....35
FIGURE 4-13: NESTED INDEX ASSIGNMENT QUANTIZERS....36
FIGURE 4-14: MODIFIED NESTED INDEX ASSIGNMENT QUANTIZERS....39
FIGURE 4-15: JPEG COMPRESSION ROUTINE....40
FIGURE 4-16: SLIDING WINDOW APPROACH....45
FIGURE 5-1: .NET ARCHITECTURE....49
FIGURE 5-2: CLASS OVERVIEW FOR THE TEST APPLICATION....51
FIGURE 5-3: TEST APPLICATION GUI....52
FIGURE 6-1: TEST IMAGES....55
FIGURE 6-2: MSSIM RECEIVING BOTH DESCRIPTIONS FROM BIRD....56
FIGURE 6-3: PSNR RECEIVING BOTH DESCRIPTIONS FROM BIRD....56
FIGURE 6-4: SDTC - SYSTEM PERFORMANCE....58
FIGURE 6-5: VISUAL RESULTS FROM SDTC....59
FIGURE 6-6: SDDDC - SYSTEM PERFORMANCE....60
FIGURE 6-7: VISUAL RESULTS FROM SDDDC....61
FIGURE 6-8: MDPCT - SYSTEM PERFORMANCE....62
FIGURE 6-9: VISUAL RESULTS FROM MDPCT....63
FIGURE 6-10: MDSQ - SYSTEM PERFORMANCE....64
FIGURE 6-11: VISUAL RESULTS FROM MDSQ....65
FIGURE 6-12: MDSQ_MOD - SYSTEM PERFORMANCE....66
FIGURE 6-13: VISUAL RESULTS FROM MDSQ_MOD....67
FIGURE 6-14: COMPRESSION AND QUALITY IN THE MD CODING SYSTEMS....68
FIGURE 6-15: COMPARISON OF ALL SYSTEM PERFORMANCES....69
FIGURE 6-16: SDDDC - DESCRIPTION ONE....71
FIGURE 6-17: MDSQ - DESCRIPTION ONE....71
FIGURE 6-18: SDDDC - DESCRIPTION ONE....72
FIGURE 6-19: MDSQ - DESCRIPTION ONE....72
FIGURE 6-20: SDDDC - DESCRIPTION ONE....73
FIGURE 6-21: MDSQ - DESCRIPTION ONE....73

List of Equations

EQUATION 4-1: RGB TO YCBCR CONVERSION 11
EQUATION 4-2: YCBCR TO RGB CONVERSION 11
EQUATION 4-3: TWO-DIMENSIONAL DISCRETE COSINE TRANSFORM 15
EQUATION 4-4: TWO-DIMENSIONAL INVERSE DISCRETE COSINE TRANSFORM 16
EQUATION 4-5: QUANTIZATION 17
EQUATION 4-6: DEQUANTIZATION 18
EQUATION 4-7: DIFFERENCE BETWEEN DC AND PRED 23
EQUATION 4-8: WAVELET DEFINITION 24
EQUATION 4-9: MDPCT TRANSFORMATION 32
EQUATION 4-10: VARIABLE C AND D CALCULATION 32
EQUATION 4-11: VARIABLE A CALCULATION 32
EQUATION 4-12: VARIABLE B CALCULATION 32
EQUATION 4-13: VARIABLE A AND B ESTIMATION FROM VARIABLE C 33
EQUATION 4-14: VARIABLE A AND B ESTIMATION FROM VARIABLE D 33
EQUATION 4-15: VARIANCES OF VARIABLE A AND B 33
EQUATION 4-16: MEAN VALUES OF VARIABLE A AND B 33
EQUATION 4-17: MAXIMUM DC VALUE 37
EQUATION 4-18: MEAN SQUARED ERROR 42
EQUATION 4-19: PEAK SIGNAL TO NOISE RATIO 42
EQUATION 4-20: Q INDEX DEFINITION 43
EQUATION 4-21: MEAN VALUES OF IMAGE SIGNALS X AND Y 43
EQUATION 4-22: VARIANCES OF IMAGE SIGNALS X AND Y 44
EQUATION 4-23: STANDARD DEVIATION OF IMAGE SIGNALS X AND Y 44
EQUATION 4-24: REWRITTEN Q INDEX DEFINITION 44
EQUATION 4-25: Q INDEX CALCULATION USING SLIDING WINDOW APPROACH 45
EQUATION 4-26: CONSTANT C_1 CALCULATION 46
EQUATION 4-27: CONSTANT C_2 CALCULATION 46
EQUATION 4-28: STRUCTURAL SIMILARITY INDEX CALCULATION 46
EQUATION 4-29: MEAN STRUCTURAL SIMILARITY INDEX CALCULATION 47

List of Tables

TABLE 1-1: QUALITY TARGETS 4
TABLE 4-1: ORIGINAL IMAGE DATA 16
TABLE 4-2: DC TRANSFORMED IMAGE DATA 16
TABLE 4-3: INVERSE DC TRANSFORMED IMAGE DATA 16
TABLE 4-4: LUMINANCE QUANTIZATION TABLE 18
TABLE 4-5: CHROMINANCE QUANTIZATION TABLE 18
TABLE 4-6: INPUT DATA 19
TABLE 4-7: QUANTIZED DATA 19
TABLE 4-8: RECONSTRUCTED DATA 19
TABLE 4-9: DIFFERENCE BETWEEN INPUT AND RECONSTRUCTED 19
TABLE 4-10: ZIGZAG INDEXING TABLE 20
TABLE 4-11: NESTED INDEX ASSIGNMENT MATRIX 36
TABLE 4-12: MODIFIED NESTED INDEX ASSIGNMENT MATRIX 38
TABLE 7-1: SUMMARY - SYSTEM PERFORMANCES 75

1 Introduction

1.1 Thesis Definition

Evaluate Multiple Description Coding as an image processing method for transferring information in error-prone networks with low transmission rate, related to quality, bit rate and file size.

The task has been given by AUC[1] who want to increase the expertise on the specialized field of MD coding. AUC is highly active within mobile- and wireless systems, where they are continuously investigating different transmission techniques. MD coding was, up until now, a relatively unknown and uncommonly used coding technique and it will be very interesting to map its opportunity sets.

1.2 Background

In today's high-tech society the transmission of images over various transmission systems is a common, and a constantly more demanded service. The enormous growth of the Internet and cellular networks over the last several years, combined with the high performance achieved by modern source coding algorithms, has resulted in the emergence of several new communication services which involve the delivery of, among other things, image data over unreliable transmission systems. However, none of these systems are without shortcomings. In most of them, packet loss and interference with other packets are common, which often results in the information transferred being heavily degraded or, at worst, useless. Another problem in these systems is transient channel shutdowns, where "channel" means distinct paths between the source and the destination. Transient shutdowns occur, for instance, because of congestion or routing delays in wired networks and deep fades, which severely limit the performance, in wireless networks. Transmission systems with a relatively high probability of such problems are often referred to as unreliable systems. Communication over such systems is often called lossy communication. This means that the received information may contain degraded information in comparison to that originally transmitted. For example, packets sent at a high rate over a fiber link are routed to a wireless network. To accommodate the lower transmission capacity of the wireless network, packets are buffered before entering the wireless network. However, if the source on the fiber link continues to send at a high rate, the buffer eventually becomes full, and packets are dropped. The problem with links such as this is that packets are dropped at random. This

---

[1] Agder University College

is illustrated in Figure 1-1.

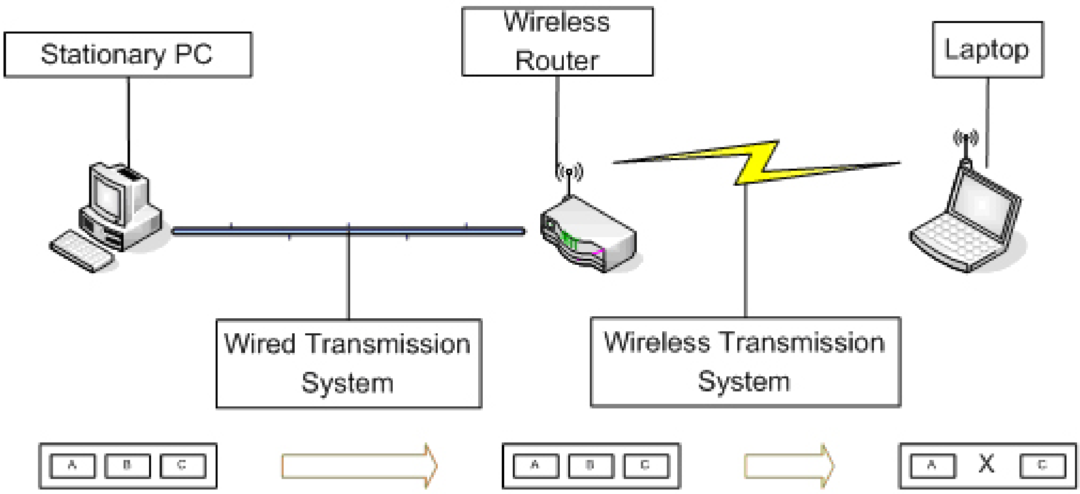


Figure 1-1: Wired vs. wireless transmission systems

The above mentioned problems can be solved by retransmitting lost and erroneous packets, which sometimes maybe more time consuming than that desired. Often, retransmissions result in the transmission being further delayed and the system being even more heavily loaded. These problems are particularly big in wireless networks, where bit errors and packet loss are more common than in wired networks. There exist several communication systems and services where retransmissions are neither possible nor desirable. For example, systems without feedback possibilities such as broadcast systems. In broadcast communication feedback messages will result in lots of traffic and should thus be avoided. Indeed, in interactive communication or real-time applications, like streaming audio and video, long delays are definitely to be avoided.

One exciting solution to these problems is to code images with the use of MD coding. MD coding introduces redundancy in the image data and then splits it up into two or more equally important streams, each with an individually lower quality than the original image. Receiving only one of these streams results in a low quality image, however, by receiving more streams the image quality is then enhanced. The different streams are correlated, so that the information in a lost stream can be more exactly estimated by using the information in the received stream(s). This technology is well suited for the transmission of data which is useful at many different levels of quality, e.g. voice, sound, image and video.

If the total amount of data representing the image becomes very large after the splitting compared to conventional representation, then the reduction of retransmission-requests are less significant. Therefore, it will be important to apply appropriate techniques for technical analysis and compression of images, prior to sending. The technical analysis of the images decides what parts of the images are important and what parts are less so, and thus which elements may be neglected.

1.3 Related Work

Since MD coding is a relatively unknown and unused coding method, there is a limited amount of literature available. Seemingly, no textbooks about MD coding exist, but there is a great deal of technical papers about previous research that is available on the Internet.

V. K. Goyal has published an introduction to different techniques within MD coding [1], while M. Pereira, M. Antonini and M. Barlaud have also written an article stating some areas of application and describe some basic coding techniques [2]. MD coding has been applied in connection with robust image coding, where V. A. Vaishampayan was the first well-known developer of image coders.

There are several different approaches of MC coding and some of them are mentioned here; Vaishampayan designed scalar quantizers [3][4], vector quantizers [5] and trellis quantizers to produce two individual descriptions. These image coders were extensions of traditional JPEG[1] coders [6]. In another approach, Y. Wang, M. T. Orchard, and A. R. Reibman developed another extension of the JPEG coder by using pairwise correlating transforms to create multiple descriptions through the introducing of a controlled amount of correlation between transform coefficients [7]. D. M. Chung and Y. Wang applied Lapped Orthogonal Transforms to construct another MD extension of JPEG coders [8]. K. Goyal, J. Kovacevic, R. Arean, and M. Vertterli made use of a more general transformation to develop yet another extension of the JPEG coder [9]. A. C. Miguel, A E. Mohr and Eve A. Riskin used SPIHT[2] to develop a scheme in a MD framework [10]. W. Jiang and A. Ortega developed a MD extension of the SPIHT coder by separating Zerotrees into polyphase components and used selective quantizing to reconstruct lost information [11].

1.4 Work Description

Noticeably, some research work about MD coding has been done earlier. The purpose of this thesis is to show that MD coding may reduce the need for, and the problems related to, retransmissions in error-prone transmission systems by a considerable scale. Based on a selection from the above-mentioned research works, some MD coding algorithms, combined with some suitable compression techniques, will be developed and implemented in a test application. This application will present and prove the effectiveness of MD coding by simulating a network image transmission where packet loss and compression rate may be determined according to the user’s demands. Seemingly, no good graphical

[1] Joint Photographic Experts Group

[2] Set Partitioning In Hierarchical Trees

applications have been developed in accordance to the earlier generated theories. This thesis will hopefully demonstrate to the reader that MD coding can produce features close to the same reconstruction quality when receiving an all-image data transmission, and a much better reconstruction quality when receiving only half the data sent, compared to the baseline systems. These constraints will hopefully be tested without an extensive amount of redundancy. Measuring image quality is a relative concept. PSNR[1] (see Section 4.4.2) is a common way of expressing reconstruction quality in image compression, but between-image comparison using PSNR is meaningless. One image with 20 dB[2] PSNR may look much better than another image with 30 dB PSNR. The level of detail in an image determines where the bound of acceptably good quality lies. In low-detail images the bounds are higher than in high-detail images. In this thesis three different images of different detail level will be used for testing. Table 1-1 shows the expected quality targets.

Table 1-1: Quality targets

Detail Level	Quality
Low	> 30 dB
Medium	> 25 dB
High	> 20 dB

The targets in Table 1-1 and the goals stated below are based on findings from other researchers [7][8][12][13]. These targets are not absolute and give merely a hint of were the expected bounds for acceptable quality lies. There is no final conclusion for what acceptable quality is. It is obvious that the quality of the reconstructed image also depends on the level of compression; when comparing the baseline systems with the MD coding systems, the same level of compression will be applied to all systems. The MD coding systems are known to add some redundancy to the image data. When receiving both descriptions, this amount of redundancy should not result in a compression rate that is 30 % lower compared to the comparison system (see Section 4.3.4) at the same quality. The drop of quality from receiving both descriptions to receiving only one should not be too extensive. A drop of quality in the range under 30 % is judged acceptable. There will be no performance of any image transmissions over physical networks since that is unnecessary in order to prove the efficiency of MD coding or to satisfy the requirements of this thesis.

[1] Peak Signal to Noise Ratio

[2] Decibel

1.5 Report outline

In the remaining chapters, a closer look at the techniques mentioned above will be presented. Chapter 2 gives a short introduction to the research methodology. Chapter 3 presents an overview and a short introduction to the developed coding systems. Chapter 4 gives a detailed introduction to the theory in this thesis. A brief introduction is given for each of the main themes before the developed and implemented techniques for compression, SD and MD coding, and quality measurement are described in detail. The application, with its design and functionality, are thoroughly described in Chapter 5. The results are presented in Chapter 6 and further discussed in Chapter 7, before a conclusion is given in Chapter 8.

2 Research Methodology

2.1 Introduction

Most research studies begin with the selection of a topic and a suitable research methodology. The selection of a research methodology is often based on the given problem, available resources, the qualifications and skills of the researcher and the target group for the research. These initial selections are very important since they will have a strong influence on, amongst other things, the structure of the study, progress, working methods, problems, results, criteria of proofs and the main goal of the study. Different types of research encompass both the theoretical basis and the methodology. There are two major research approaches, namely quantitative research and qualitative research, it is also relatively common to make use of both approaches in a mixed research method approach [14]. For instance, information from a qualitative research may be an important supplement to the information collected by means of a quantitative approach.

2.2 Qualitative and Quantitative Research

In qualitative research the goal is to understand a social or human problem from different perspectives. Such research is often conducted in a natural setting and is primarily characterized by a common objective setting of developing models and theories which help us understand social phenomena as they naturally appear (that is, as opposed to experimental situations). Bringing about the meaning which the participants themselves add to the phenomenon is of great importance, and also their experiences and views. The researcher often performs fieldwork, without any preconceived views, with a goal to understand the phenomenon of interest, how this phenomenon varies at different circumstances and why. In qualitative research, the researcher, participants, and the audience interpreting the results, are included in the study. The researcher cooperates with those being studied and actively works to minimize the distance between them. To improve accuracy, participants are often asked to verify the data to ensure it matches their impression of the information they believe they gave to the researcher. It is common that qualitative research is used to generate theories rather than test them [15].

In quantitative research, on the other hand, it is often assumed that the reality is objective and independent of the researcher, and therefore can be studied objectively. The research is primarily based on deductive and hypothetical forms of logic, theories and hypotheses are often composed at an early stage [16] and, further, they are often tested in a cause-effect order. The goal is to develop generalizations that contribute to the existing theory to enable the researcher to predict, explain and understand behavior [15]. From this, quantitative research is said to be some additional research on an earlier identified problem which is based on testing of theories, different types of measurements and analyses by means of statistical techniques. The goal is often to determine whether the predictive generalization of a theory holds true [17].

Separately, these two methods of research are very different in their underlying assumptions and the procedures used for data collection and analyses.

2.3 Summary

This thesis evaluates a problem that has been previously identified. The researchers are distant and nearly independent of the topics that have been researched. The work will consist of developing, implementing and testing different MD coding and measuring techniques. The goal is to give a contribution to the existing theory by proving the effectiveness of certain MD coding techniques, compared with some conventional baseline systems. Some hypotheses were composed at an early stage (in the proposal) and later tested in a cause-effect order. In order to achieve a satisfactory result in this thesis, knowledge of formal theory was required and the developed models are relatively simple and precise. All these factors indicate a quantitative approach should be used.

From this, and according to [16], it can now be concluded that the researchers in this thesis make use of an “expository” research method, a type of method which often is quantitative in character. However, since the researchers also make use of some qualitative research, e.g. in the form of subjective result analysis, the final conclusion is that this research work is based on a mixed research method.

3 The System

3.1 Introduction

To reach the goal about converting an image into MD coded data suitable for simulated image network transmission, the data have to go through several operations.

3.2 System Overview

The developed systems consist of an encoder and a decoder and Figure 3-1 gives a high-level overview of these systems.

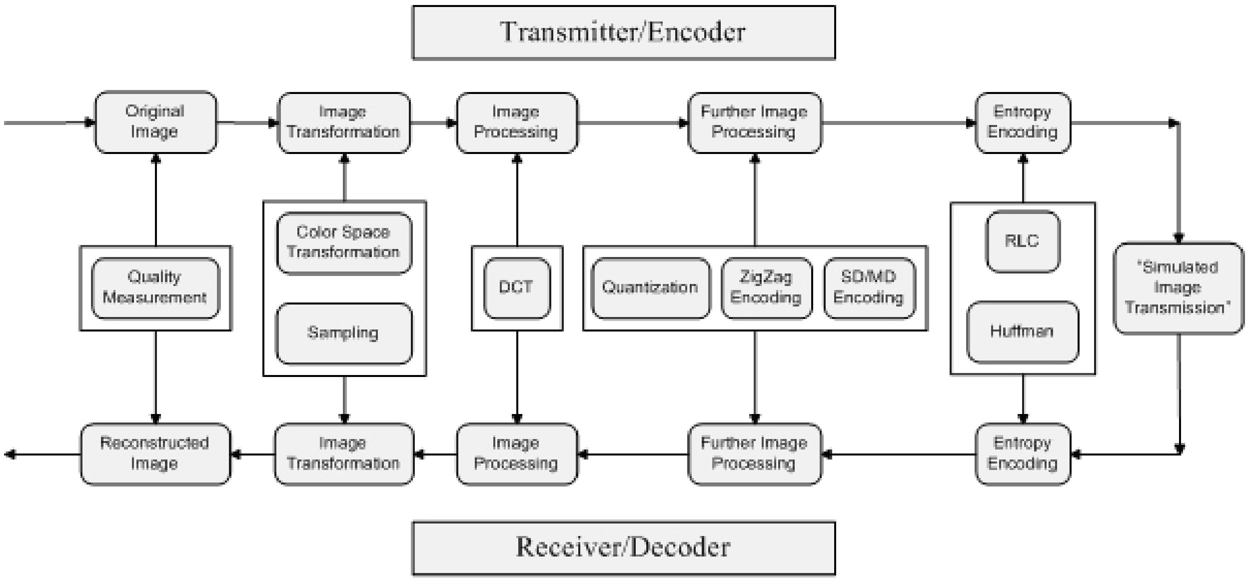


Figure 3-1: System overview

The order of the different functions in the *Further Image Processing* section may vary a bit from system to system. Each system is described in detail in chapter 4. The encoder works as follows; an image of type bmp[1], jpeg, gif[2] or png[3] is loaded into memory and converted into a bitmap. The RGB[4] values are then converted into the YC_bC_r color space (see

[1] Basic Multilingual Plane/Bitmap

[2] Graphics Interchange Format

[3] Portable Network Graphics

[4] Red, Green and Blue

Section 4.2.2) before the chrominance components (C_b and C_r) are downsampled to a quarter of their original size (see Section 4.2.3). Then, the Y, C_b and C_r components are transformed using DCT[1] (see Section 4.2.4) before the transformed coefficients are quantized using a standard JPEG quantization matrix (see Section 4.2.5). The quantized output matrix is then, for simplicity, converted into a one-dimensional matrix using a Zigzag coding algorithm (see Section 4.2.6). Then, the signal (image) is split into two individual descriptions (basically bit streams) by using a selected SD or MD coding algorithm (see Section 4.3). After the SD or MD coding the individual descriptions are further compressed by RLC[2] (see Section 4.2.7) and Huffman coding (see Section 4.2.8), respectively. The output of the Huffman coding is two separate byte arrays.

When all coding functions are accomplished, a simulated image transmission, where packet loss may be embedded, is performed. In this relation, packet loss corresponds to loss of one description.

After transmission, the inverse coding functions are, in reverse order, applied to all successfully transmitted data before the image data are reconstructed. The reconstructed image is then compared to the original image, using two different quality measurement techniques, and the reconstructed image is finally displayed.

[1] Discrete Cosine Transform

[2] Run Length Coding

4 Theory

4.1 Introduction

This chapter describes the theory around the main themes in our thesis, namely compression, MD coding and quality measurement.

Even though the three color components (Y, C_b and C_r), in the description of each single procedure within compression, SD and MD coding, and quality measurement, are referred to as one component, it is important to mention that all the procedures are performed separately on all the three color layers. This makes the project far more complicated than, e.g., if all procedures were performed on black and white images. Thus, the chrominance components (C_b and C_r) could be neglected. This means that each step in the systems (see Section 3.2, Figure 3-1) is performed on each color layer.

4.2 Compression

4.2.1 Introduction

Uncompressed signals (i.e. images, audio and video) require large storage capacity and transmission bandwidth. Despite the continuous progress within development of storage components, processor speeds and transmission technologies, the recent growth of i.e. multimedia based web applications will constantly outstrip the capacity of the available technology. This progress has both emphasized the need for effective signal coding techniques and made signal compression central in today's storage and transmission technology. Accordingly, in order to evaluate the real effectiveness of the MD coding techniques, it is absolutely necessary to compress the image data to a size comparable to that of conventional compression algorithms. MD coding without any compression is probably a useless technique because a compressed image then might be transmitted several times sending the same amount of data.

Two alternative compression algorithms were considered for this thesis, namely wavelets and DCT, and finally the DCT algorithm was chosen. In Section 4.2.4, a detailed description of the DCT algorithm is given, together with the additional procedures required to make the compression as good as possible. These additional procedures are; color conversion, downsampling of color components, Zigzag coding, quantization, Run Length coding and Huffman coding. A short description about the inverse functions of the compression procedures mentioned above is also described. At the end of the section a brief description of the wavelet technology, and also a basis for the DCT choice is given.

4.2.2 Color Space Transformation

A digital picture often consists of three layers of equal size. These layers correspond to the red, green and blue color information for each pixel in an image. Such an image is said to be in the RGB color space. The RGB color space is convenient for interpreting and displaying images, but it does not make use of the correlation between color components and low sensitivity of the human eyes to chrominance components. Therefore, when it comes to compression, it is appropriate to convert the image data into a suitable color representation, such as $\mathrm{YC_bC_r}$ [18]. The Y component is luminance and stores the intensity of each pixel (actually gray scale). The $\mathrm{C_b}$ and $\mathrm{C_r}$ component are blue and red chrominance, respectively, and contain the color information for each pixel. The green chrominance value is not stored, because with two chrominance values it is a simple calculation to get the third [19]. The conversion can be done using the following equations [20]:

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

Equation 4-1: RGB to YCbCr conversion

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & -0.001 & 1.402 \\ 1.000 & -0.344 & -0.714 \\ 1.000 & 1.772 & 0.001 \end{bmatrix} \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix}$$

Equation 4-2: YCbCr to RGB conversion

4.2.3 Downsampling of Color Components

Downsampling may reduce the image size by one-half or one-third, with nearly no impact on perceived image quality. This is done by dividing the pixels into blocks before calculating the average value for each block. Only this average value is used to represent that whole group. This is done only to the chrominance (colored) components, thus this is not necessary for gray scale images [21][22].

If an image is broken into blocks of 2×2 pixels (see Figure 4-1), the compressed chrominance layers are 1/4 of their original size, resulting in an overall reduction in image size by one-half. Explanation: Because each image layer accounts 1/3 of the image data, the image size reduction is given by:

$$\text{New image size} = \frac{1}{3} + \frac{1}{4} \times \left(\frac{1}{3} + \frac{1}{3} \right) = \frac{1}{3} + \frac{2}{12} = \frac{1}{2} \text{ original image size}$$

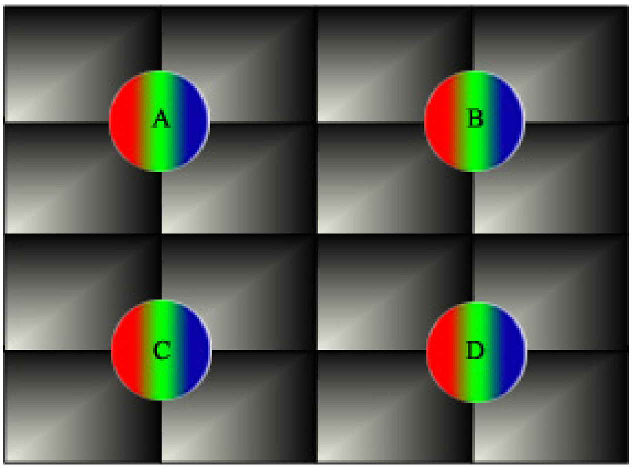


Figure 4-1: Downsampling of chrominance components

In Figure 4-1 the black and white squares represent pixels, while the RGB colored circles represent the downsampled chrominance data. Downsampling is not absolutely necessary as the DCT algorithm actually works on each independent component and does not care what data it is. But it does increase the compression ratio as it removes unnecessary information in the chrominance components without the human eye detecting the difference. The reason for this is that the human eye is not as sensitive to high-frequency color information as it is to high-frequency luminance.

Upsampling

Before the reconstructed (transmitted) image can be displayed, the chrominance components have to be upsampled to their original size. There are several ways to perform this operation. First, a simple way of upsampling, by spreading the downsampled values to the four original components around, was chosen. Figure 4-2 illustrates the outcome of the upsampling procedure.

A	A	B	B
A	A	B	B
C	C	D	D
C	C	D	D

Figure 4-2: Upsampling of chrominance components

The procedure described above gives an acceptable, but not optimal, result. Therefore, another, and better, procedure, which introduces a sort of weighting of the downsampled values nearby each pixel, was implemented. This procedure is illustrated below.

A	0.75A+0.25B	0.25A+0.75B	B
A	0.75A+0.25B	0.25A+0.75B	B
C	0.75C+0.25D	0.25C+0.75D	D
C	0.75C+0.25D	0.25C+0.75D	D

A	A	B	B
0.75A+0.25C	0.75A+0.25C	0.75B+0.25D	0.75B+0.25D
0.25A+0.75C	0.25A+0.75C	0.25B+0.75D	0.25B+0.75D
C	C	D	D

Figure 4-3: Weighted upsampling of chrominance components

In Figure 4-3, the chrominance components located in the corners are estimated from the values of the nearest downsampled component. The remaining chrominance values are then estimated by interpolating the downsampled values. This means that each original chrominance value is estimated by adding 75 percent of the nearest downsampled value and 25 percent of the second nearest downsampled value. To obtain as good reproduction of the original chrominance layers as possible, this interpolation is done both horizontally (Figure 4-3 left) and vertically (Figure 4-3 right) before an average value is calculated for each pair of correspondent components [22].

4.2.4 Discrete Cosine Transform

DCT is a lossless algorithm, which means that the reconstructed image contains no degradations with respect to the original image. One disadvantage with lossless algorithms, compared to lossy algorithms, is that they offer much lower compression ratios.

DCT is closely related to the FFT[1], a technique for converting a signal into elementary frequency components. The frequency domain is a better representation for the data because it makes it possible to separate out, and discard, unimportant information to human perception. The human eye is not very sensitive to high frequency changes, especially in photographic images [23]. Accordingly, the high frequency components are less important than the low frequency components and can, to some extent, be discarded.

An image is divided into $N \times N$ pixel blocks, typical 8×8 blocks, and the two-dimensional DCT is then applied to each block. In this project, a block size of 8 (N = 8) pixels was used. This block size is considered as an optimal size for image compression, following the JPEG standard [24].

The DCT is given by [23][24]:

$$T(i,j) = \frac{2}{N} C(i) C(j) \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} S(m,n) \cos\left[\frac{\pi(2m+1)i}{2N}\right] \cos\left[\frac{\pi(2n+1)j}{2N}\right]$$

$$\text{where the coefficients } C(k) = \begin{cases} \frac{1}{\sqrt{2}} & (k=0) \\ 1 & (Otherwise) \end{cases}$$

Equation 4-3: Two-dimensional Discrete Cosine Transform

In Equation 4-3, S is the array of $N \times N$ original values and T is the array of $N \times N$ transformed values.

The result of the transformation is an $N \times N$ block containing the frequency domain representation of the original $N \times N$ block of image data. In each block, the low frequency components are located in the upper-left corner, and the high frequency components are located in the bottom-right corner. The upper-left value in each block is called the DC coefficient, containing the average of the original pixel values within the block, and the other values are called AC coefficients. To convert the image back to its origin, the IDCT[2] is used. The IDCT is given by [24]:

[1] Fast Fourier Transform

[2] Inverse Discrete Cosine Transform

$$S(m,n)=\frac{2}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}C(i)C(j)T(i,j)\cos\left[\frac{\pi(2m+1)i}{2N}\right]\cos\left[\frac{\pi(2n+1)j}{2N}\right]$$

$$\text{where } C(k)=\begin{cases}\frac{1}{\sqrt{2}} & (k=0)\\ 1 & (Otherwise)\end{cases}$$

Equation 4-4: Two-dimensional Inverse Discrete Cosine Transform

The tables below illustrate how a block of image data is transformed by the DCT and then inverse transformed by the IDCT, to achieve the original image data.

Table 4-1: Original image data

140	144	147	140	140	155	179	175
144	152	140	147	140	148	167	179
152	155	136	167	163	162	152	172
168	145	156	160	152	155	136	160
162	148	156	148	140	136	147	162
147	167	140	155	155	140	136	162
136	156	123	167	162	144	140	147
148	155	136	155	152	147	147	136

Table 4-2: DC transformed Image Data

186	-18	15	-9	23	-9	-14	19
21	-34	26	-9	-11	11	14	7
-10	-24	-2	6	-18	3	-20	-1
-8	-5	14	-15	-8	-3	-3	8
-3	10	8	1	-11	18	18	15
4	-2	-18	8	8	-4	1	-7
9	1	-3	4	-1	-7	-1	-2
0	-8	-2	2	1	4	-6	0

Table 4-3: Inverse DC transformed image data

140	144	147	140	140	155	179	175
144	152	140	147	140	148	167	179
152	155	136	167	163	162	152	172
168	145	156	160	152	155	136	160
162	148	156	148	140	136	147	162
147	167	140	155	155	140	136	162
136	156	123	167	162	144	140	147
148	155	136	155	152	147	147	136

4.2.5 Quantization

Quantization is used to reduce the number of bits needed to store the transformed DCT coefficients by reducing the precision of these values. The purpose of this operation is to achieve further compression by representing DCT coefficients with no greater precision than what is necessary to achieve the desired image quality. Quantization is a lossy process and, since it makes a good trade-off between compression rate (bit rate) and image quality, one of the most important compression functions in an encoder.

The DCT procedure resulted in a two-dimensional matrix containing DCT coefficients, where low frequency components are located at the upper-left and increasingly high frequency components are located at the lower-right in the matrix, for each block. Then the precision of the coefficients can be reduced more and more as we move away from the DC coefficient (towards bottom-right). Thus, each data block in the two-dimensional DCT matrix is quantized in conjunction with a carefully designed quantization table. This process can be expressed as follows [24]:

$$F^{Q}(y,x) = IntegerRound\left(\frac{F(x,y)}{Q(x,y)}\right)$$

Equation 4-5: Quantization

In Equation 4-5, x and y give the position of each element. Thus, each element $F(x,y)$ from the DCT matrix is divided by the corresponding element $Q(x,y)$ from the quantization matrix, and the result is rounded to the nearest integer. Thus $F^{Q}(x,y)$ denotes the quantized DCT coefficients, normalized by their individual quantizer step size. The quantizer step size for each coefficient $F(y,x)$ is the value of the corresponding element $Q(x,y)$ from the quantization matrix. The most important coefficients for reconstruction of images (those closer to the upper-left corner) are encoded with a small step size, while coefficients that are less important (those closer to the lower-right corner) are encoded with larger step sizes. The quantization table is a set of 64 quantization values which is chosen to reduce the precision of each coefficient to no more than necessary.

Because of the nature of the MDPCT[1] system (see Section 4.3.3), the Zigzag procedure is performed before quantization. Quantization is performed on the one-dimensional output matrices of the Zigzag

[1] Multiple Description by Pairwise Correlating Transform

procedure. The two-dimensional luminance and chrominance quantization matrices are therefore reduced to two one-dimensional matrices. The new one-dimensional matrices consist of every second coefficient in the original matrices, and are thus half the length of this. The reason for this is further explained in the section about MDPCT (see Section 4.3.3).

In this thesis, the standard JPEG quantization tables for both luminance and chrominance components [25] were chosen and they are shown in Table 4-4 and 4-5, respectively.

Table 4-4: Luminance quantization table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Table 4-5: Chrominance quantization table

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Dequantization

Dequantization is the inverse function of quantization and implies removal of the above-mentioned normalization. By multiplying the quantized values with the corresponding elements from the quantization matrix, it is possible to estimate the original unquantized values, which are suitable for the IDCT procedure. This process is given by [24]:

$$F^{Q'}(x,y) = F^{Q}(x,y) \times Q(x,y)$$

Equation 4-6: Dequantization

In Equation 4-6, $F^{Q'}(x,y)$ denotes the estimated coefficients while $F^{Q}(x,y)$ and $Q(x,y)$ denotes the quantized coefficients and the corresponding values from the quantizer matrix, respectively.

Tables 4-6 to 4-9 on the next page give an example of the described quantization and dequantization procedures. A data block containing luminance data is quantized and dequantized with the quantization matrix given in Table 4-4. The example also illustrates why the quantization procedure is called a lossy compression process.

Table 4-6: Input data

330.6	-3.0	-12.9	-5.2	1.9	-1.7	-2.7	1.3
-24.3	-12.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-11.7	-9.1	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-1.6	2.6	-3.5	-2.0	1.7	1.1	-0.5	-0.3

Table 4-7: Quantized data

21	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Table 4-8: Reconstructed data

336	0	-10	0	0	0	0	0
-24	-12	0	0	0	0	0	0
-14	-13	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Table 4-9: Difference between input and reconstructed data

-5.4	-3.0	-2.9	-5.2	1.9	-1.7	-2.7	1.3
-0.3	-0.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
2.3	3.9	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-1.6	2.6	-3.5	-2.0	1.7	1.1	-0.5	-0.3

The quantization and dequantization procedures take the following steps in Tables 4-6 to 4-9:

- An encoder input value, i.e. 330.6 in the upper left corner is scaled by the corresponding step size (16) from the luminance quantization table: $330.6/16 = \underline{20.6625}$

- The scaled value is rounded to the nearest integer, resulting in quantizer index $= \underline{21}$

- The quantizer index is multiplied with the corresponding quantization step size: $21 \times 16 = \underline{336}$

- The difference (or error) between the original input value and the output data is then: $330.6 - 336 = \underline{-5.4}$

Since the difference between the input and output data is different from zero, the quantization is called a lossy compression process.

4.2.6 Zigzag Coding

The output of the quantization procedure is a two-dimensional matrix containing quantized DCT coefficients, where components of low frequency are located in the upper left corner and increasingly high frequency components are located toward the bottom right, for each block. In the MDPCT system, the Zigzag coding is performed before the quantization.

To simplify SD and MD coding, RLC and Huffman coding, it was desirable to convert the two-dimensional DCT matrix into an ordered one-dimensional Zigzag array, sorted from lowest to highest frequency values. Since low frequency components are more important than high frequency components for proper reconstruction of the image, the Zigzag procedure helps us to determine the most important coefficients, and thus which require extra redundancy.

The Zigzag procedure was performed by using the Zigzag indexing table shown in table 4-7 [26]. This is for convenience sake displayed as a two-dimensional table, but is in reality a one-dimensional table of length 64. The start of the table is in the upper-left corner, and the end in the bottom-right corner. It runs from left to right in each row, and from top and down.

Table 4-10: Zigzag indexing table

0	1	7	8	-7	-7	1	7
7	7	8	-7	-7	-7	-7	1
7	7	7	7	7	8	-7	-7
-7	-7	-7	-7	1	7	7	7
7	7	7	7	1	-7	-7	-7
-7	-7	8	7	7	7	7	7
7	1	-7	-7	-7	-7	8	7
7	7	1	-7	-7	8	7	1

The formula for placing the DCT coefficients in a one-dimensional array is as follows;

```
for (int i=0; i<64; i++)
{
      S1pointer += index[i];
      a = S1pointer/8;
      b = S1pointer%8;
      zz[i] = S1[a,b];
}
```

where *S1* is the array containing the DCT coefficients, and the *zz* array is the new array containing the ordered coefficients. The index array is the Zigzag indexing table. The *S1pointer* adds the next value in the index table to itself for each increasing *i*. The variables *a* and *b* represents the rows and columns in the DCT matrix, respectively, and are calculated on the basis of *S1pointer*.

The Zigzag procedure is visually illustrates below. Figure 4-4 illustrates an example of how the Zigzag procedure works on a 8×8 pixels image. In this thesis the block size equals 8, but for simplicity a block size equal to 4 was chosen in this example. The yellow squares denote the DC coefficients and the white squares denote the AC coefficients. The numbers indicates the output order of the Zigzag procedure.

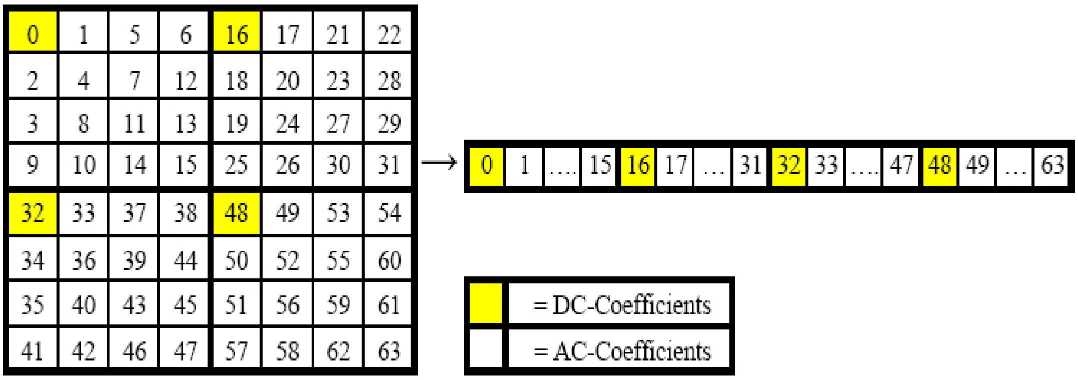


Figure 4-4: Input (left) and output (right) data of the Zigzag procedure

Inverse Zigzag coding

The Inverse Zigzag scan is performed by reversing the Zigzag table, and repeating the inverse scan for each block of the Zigzag array.

4.2.7 Run Length Coding

Run Length Coding is applied to the AC components of the zigzag-scanned blocks. In MD coding, each individual description is Run Length Coded. After quantization the one-dimensional matrix will consist of lots of zeroes [25], and it would present a great compression gain from not coding all the zeroes individually. Thus a combination of run-amplitude is Huffman encoded. The codeword consists of the number of zeroes before a non-zero coefficient and the size (number of bits) needed to represent that coefficient (run, size). The codeword is followed by the additional bits which define the coefficient amplitude. If the amplitude is negative, all the bits which define the amplitude are inverted. This gives a 0 in the sign bit of the amplitude, which indicates a negative value for decoding (see Figure 4-5). This run-amplitude coding is efficient because there is a strong correlation between the number of zeroes and the following non-zero amplitude. A long run of zeroes is usually followed by small amplitude, and a short run of zeroes is usually followed by larger amplitude [27]. In order to keep the code table small, a maximum run of 15 zeroes is encoded and size is maximum 11 (see Appendix A [25]). Two extra codes are used, EOB[1] and ZLR[2]. EOB is coded if the rest of the coefficients in the block are zero, and ZLR is coded if run is bigger than 15. A run of 19 zeroes followed by 4 is encoded as (15, 0) (3, 3) 100, where (15, 0) = ZLR. Figure 4-5 gives an example of a Run Length Coded signal.

Input:	4 2 0 0 0 0 6 -1 0 0 0 1 2 0 0 0 0
Output:	(0,3)100 (0,2)10 (4,3)110 (0,1)0 (3,1)1 (0,2)10 EOB

Figure 4-5: Run Length Coding

[1] End Of Block

[2] Zero Run Length

4.2.8 Entropy Coding

Entropy coding is used on each of the two individual descriptions for further compression of the image data prior to transmission. The most commonly used techniques for entropy coding are Huffman coding and arithmetic coding. Arithmetic coding is mathematically superior to Huffman coding [28], but it is covered by patents, slower to code and decode than Huffman coding, and it does not provide much net space gain. The Huffman algorithm, on the other hand, is comparatively easy to implement and produces a lossless compression of images. Thus, the Huffman coding was chosen as the entropy coding technique, in preference to Arithmetic coding. The Huffman tables contain variable-length prefix-free code words for encoding a source symbol. Prefix-free means that the bit string representing some particular symbol is never a prefix of the bit string representing any other symbol [29]. The length of the code words depends on the probability of the corresponding symbol. If a symbol appears frequently, the code word should be shorter than for a symbol which appears less frequently. Calculating Huffman tables for each image gives a small amount of compression gain over using some default Huffman tables [27]. In this thesis the Huffman tables given in [25] (see appendix A) was used for the Huffman coding. Two procedures are used, one for the DC coefficient and the other for the AC coefficients. The DC coefficient is always encoded first, followed by the AC coefficients. The DC coefficients are coded differentially. This means that the DC coefficient is subtracted from the most recently coded DC coefficient (PRED). The difference, given by Equation 4-7, is coded.

$$DIFF = DC - PRED$$

Equation 4-7: Difference between DC and PRED

The AC coefficients are first run length coded and then Huffman coded. Each (run, size) pair from the output in Figure 4-5 is converted to its appurtenant codeword (bit string).

4.2.9 Wavelet Transform versus DCT

Wavelets are functions which are defined over a finite interval with mean value of zero. A simple and good introduction to the wavelets theory is found in [30]. The basic idea of the wavelet transform is to represent any arbitrary function as a set of given wavelets, or basis functions [31]. These basis functions are obtained from a single prototype wavelet, called a mother wavelet, by scaling and shifts. Wavelet transformations can be divided into two groups; CWT[1] and DWT[2], where the latter is used within image compression. According to [31], a wavelet ψ can be defined as follows:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)$$

Equation 4-8: Wavelet definition

In Equation 4-8, *a* denotes the scaling, *b* denotes the shift and t denotes the time.

During the last several years, wavelet transformations have achieved a widespread acceptance, particularly within image compression research. Since wavelet functions are randomly chosen, this technique is more flexible than DCT, which is a static function that can not be adapted to the source data. Wavelets are also chosen as basis functions in JPEG 2000 [32]. There are several reasons to this positive progress, and one of the most important reasons is obviously that wavelet coding (often called subband coding) has outperformed other coding schemes like the one based on DCT, in many applications. Since there is no need to split the input image into blocks and its basis functions have variable length, wavelet coding schemes at higher compression avoid some annoying blocking artifacts, which may occur in DCTs. The transformation of the whole image also introduces inherent scaling. In addition, wavelet transformations give a good localization in time and spatial frequency domain, they are robust against transmission and decoding errors and they are well matched to the HVS[3] characteristics (see Section 4.4.1). Many of these qualities simplify the image transmission process.

However, the DCT compression procedure also has several important advantages. It is relatively easy to implement, it has a satisfactory performance and it has some special purpose hardware available for implementation. According to [33], DCT compression is optimized for visual perception and, therefore, it may be more visually correct than

[1] Continuous Wavelet Transform

[2] Discrete Wavelet Transform

[3] Human Visual System

wavelet transformations, despite that wavelet transformations are probably more mathematically correct. Unfortunately, there are also several disadvantages following the DCT procedure [31]. One of them is the above-mentioned visual blocking artifacts. Since the input images are split into a number of blocks and all the blocks are transformed separately, the boundaries between the blocks often become visible. Therefore, it is impossible to completely decorrelate the blocks at their boundaries using DCT. Another disadvantage is that DCT does not perform efficiently for binary images characterized by large periods of constant amplitude, followed by brief periods of sharp transitions.

Wavelets also have several disadvantages [31]. The cost of computing DWT, compared to DCT, may be higher. By use of large DWT basis functions or wavelet filters, a blurring and ringing noise may occur near edge regions in images. Further, wavelet transformations usually give lower quality than DCT at low compression rates, which means that compression with DCT may achieve a higher compression rate than wavelets with less loss of visual details.

Despite some great advantages of DWT over DCT, DCT was chosen as the data transformation technique, within image compression, in this thesis. One reason to this choice is that DCT is a well-known, and frequently used, technique which satisfies the requirements to compression of regular images, like photos. Another important reason is that compression is just a sub-theme of the thesis and since wavelet transformations are considerably more complex processes than DCT transformations, the compression section of the thesis could have overshadowed the main theme, MD coding, by use of wavelet based compression. JPEG, which uses DCT, is also the most widely used image format [34], and the comparison system in this thesis is based upon JPEG. The use of DCT ensures that the results in this thesis are a consequence of MD coding, and not simply the effect of using Wavelet versus DCT.

4.3 Coding Systems

4.3.1 Introduction

This section describes the two baseline systems, SDTC[1] and SDDDC[2], the three MD coding systems, MDPCT, MDSQ[3] and MDSQ_MOD[4], and the system for quality comparison, the JPEG system, developed in this thesis.

MD coding can be applied when lossy communication is tolerated [35]. That is, when the users are satisfied with a degraded version of the original signal. Examples of data types which allow lossy compression are audio, video, speech and images. However, there might be exceptions. For instance, images in medicine should not be compressed lossily if they contain important details. MD coding can also be applied in different types of transmission systems, but when we hear about communication by use of MD coding, it is usually meant communication in packet network. The data can be organized in different streams of packets, where one stream corresponds to one description.

MD coding is based on the diversity principle [12]. Diversity is a well-known technique which is commonly used to improve the reliability in communication systems, especially within mobile radio communication [36]. This is done by sending data over multiple channels, different paths through the network, each with independent error events. Then, the probability of receiving some data from at least one channel is highly increased. A MD coder generates two or more descriptions with equal importance from a single source signal (an image in this case), and it is assumed that the probability of loss of all descriptions during transmission is very low. Traditionally the source signal is split into two descriptions where each description has a certain probability of loss or exposure to errors during transmission.

In this thesis only the two-description case is considered, where it is required that a high-quality reconstruction of the original image can be decoded from both descriptions together, while a lower, but still acceptable, reconstruction quality can be decoded from one of the two individual descriptions. Finally, if no descriptions are received (very low probability) it is impossible to reconstruct the original image without performing a retransmission. To ensure that the quality of the reconstructed image from one single description is acceptably high, each description needs to carry sufficient information about the original image. To manage this, a certain degree of correlation has to be embedded

[1] Single Description Transform Coding

[2] Single Description Double DC

[3] Multiple Description Scalar Quantization

[4] Multiple Description Scalar Quantization Modified

between the individual descriptions. Without this correlation, it is impossible to make an estimate of one description from another and, therefore, this correlation is a requirement to all MD coding systems. However, the correlation will reduce the coding efficiency compared to conventional SD coding [7]. The main challenge in development of MD coders is to make the descriptions individually good but not too similar.

4.3.2 Baseline Systems

The two baseline systems, called *Single Description Transform Coding* and *Single Description Double DC,* were developed to demonstrate the need for a new coding technique, namely the MD coding. Both baseline systems are based on the principle in [1] and a system overview is illustrated in Figure 4-6.

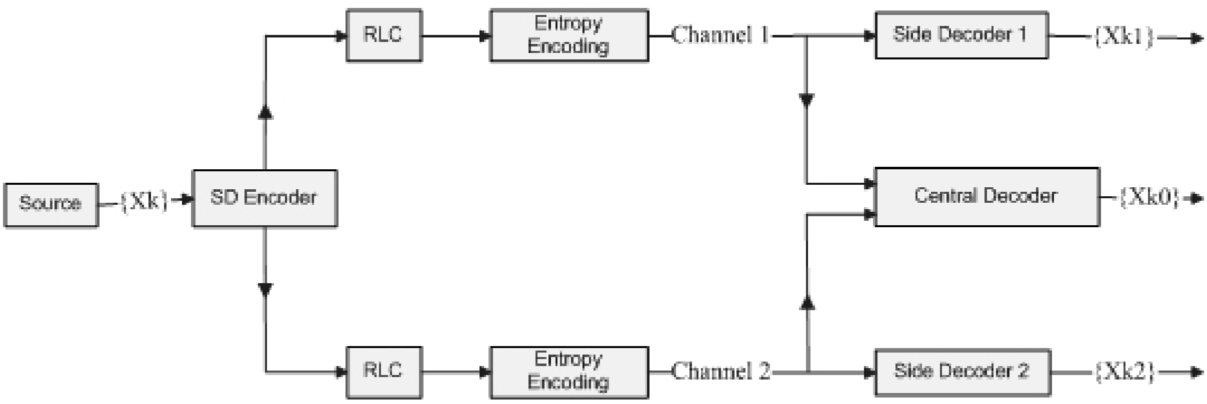


Figure 4-6: Baseline system overview

The output data $\{X_k\}$ from a *source* is split into two descriptions by means of different algorithms in the *SD Encoder*. The two descriptions are individually Run Length and Huffman encoded before they are transmitted over two separate channels, *Channel 1* and *Channel 2*. If the data on both channels are transmitted successfully the *Central Decoder* is used to decode the transmitted data. On the other hand, if the data on either *Channel 1* or *Channel 2* are lost during transmission, the *Central Decoder* is useless since half of the data needed to decode the transmitted data $\{X_k\}$ are lost. Then the *Side Decoder 1* or the *Side Decoder 2* (the one which belongs to the received description) is used to decode the received data.

Both baseline systems use the one-dimensional output array from the Zigzag procedure as their input parameter, but the data sent on each channel is different in the two systems.

Single Description Transform Coding

In the first system the incoming array of coefficients is split into two descriptions by sending odd indexed blocks on channel one and even indexed blocks on channel two. The block size is set to 16. This is because the downsampling procedure requires that four 8×8 blocks is downsampled to one 8×8 block. So each block consists of six 8×8 blocks after downsampling, namely four Y blocks, one C_b block and one C_r block. This means that the two descriptions are independent of each other and carry different information. Figure 4-7 illustrates this splitting.

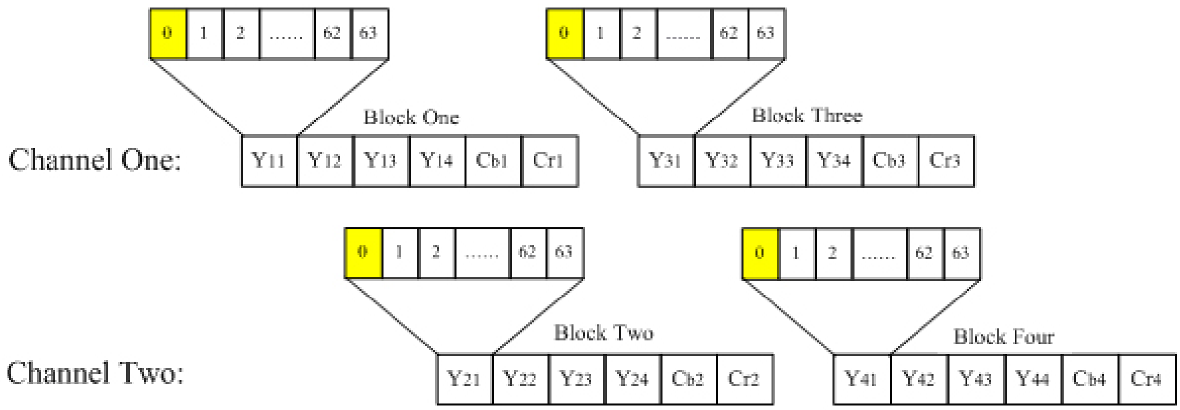


Figure 4-7: Illustration of the SDTC splitting

This system does not contain any function for estimation of lost descriptions. If only one description is successfully transmitted, the pixel values in the lost blocks are set to black. By loss of one description the quality of the reconstructed image becomes very poor compared to the original image, and the receiver will see a disturbing checkerboard pattern or stripes (depending on the image width) on the received image.

Obviously, this system is not able to handle packet loss in an acceptable way. The system relies on a retransmission function in order to produce a reconstruction of the original image with a satisfactory quality. Therefore, a system which guarantees a better quality on the transmitted data by loss of one description was developed.

Single Description Double DC

The second system is an improvement of the SDTC system. Here, all the DC coefficients are duplicated and transmitted on each channel while the AC coefficients are split on the basis of their index. The DC coefficient and all odd indexed AC coefficients are sent on channel one, while the DC coefficient and all even indexed AC coefficients are sent on channel two (see Figure 4-8). This means that the two descriptions still are independent but now they are carrying a small amount of similar data. Neither this system contains any function for estimation of lost descriptions, but the duplicating of DC coefficients gives us an opportunity to reconstruct the original image with a considerable better quality than by use of SDTC. This splitting is illustrated in Figure 4-8.

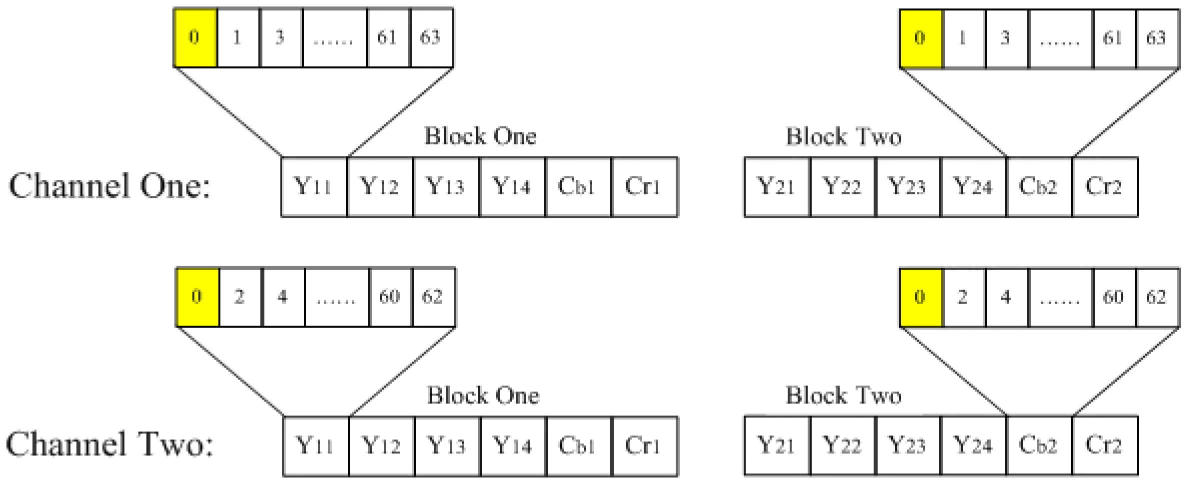


Figure 4-8: Illustration of the SDDDC splitting

By loss of one description the AC values in the lost blocks are reconstructed by assigning each coefficient a zero value. By receipt of both descriptions the receiver gets two similar sets of DC coefficients, where one of the sets then will be discarded as redundant data. However, this problem can be neglected since the DC coefficients in a transformed image only constitute a small amount of data compared to the total amount of image data. Duplicating the DC coefficients will result in a lower coding efficiency in this coder than in the SDTC coder. But seeing that this system guarantees a much better quality of the reconstructed image, by loss of information, this is still a great improvement from the first system. Unfortunately, the quality is still unacceptably poor. The system is not able to produce a satisfactory reconstruction of the original image, by loss of one description, and there is still a great need for retransmissions.

Baseline Summary

Despite the above-mentioned signal splitting, both the baseline systems are called SD coding systems. The SDTC system is called a SD coding system because it does not introduce any redundancy to the original data amount. There is neither introduced any degree of correlation between the two generated descriptions. The only reason to split the data into two descriptions is to make it possible to simulate packet loss during the simulated network image transmission. In the SDDDC system a small amount of redundancy is introduced, but no degree of correlation. Therefore, neither this system can be called a MD coding system.

Some error concealment techniques which may improve the quality of the reconstructed images in systems like SDTC and SDDDC exist, but since this is beyond the scope of this project it will not be examined more closely in this paper.

4.3.3 Multiple Description Coding Systems

There are several ways for generation of “multiple descriptions” and several factors to take into consideration when designing MD coders. The main challenge is to find out how to achieve a good trade-off between the coding efficiency and the quality on the reconstructed image from only one description.

Multiple Description by Pairwise Correlating Transform

The first MD coding system developed in this thesis is based on the principle illustrated in Figure 4-9.

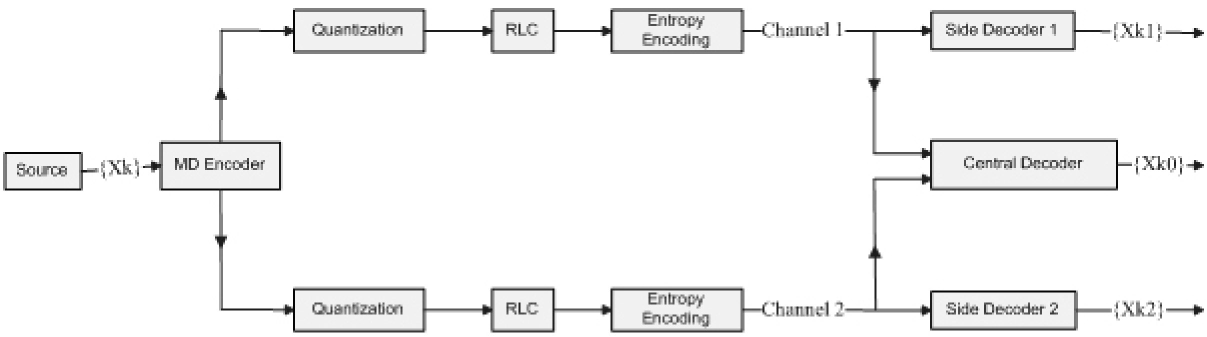


Figure 4-9: MDPCT overview

This system has almost the same structure as the baseline systems. The only difference is that zigzag coding is performed before MD coding, and quantization is performed on each individual description subsequent to the MD coding.

This system was developed to solve the problems in the baseline systems by coding pairs of variables. Here, “variable” means a collection of DCT coefficients. In the baseline systems two independent variables were sent over separate channels. In the MDPCT system, a controlled amount of correlation between two variables, *A* and *B*, is introduced by transforming them into two new variables, *C* and *D*, as illustrated in Figure 4-10.

Figure 4-10: Illustration of the MDPCT transformation

If there are N DCT coefficients, each of the four variables (*A, B, C* and *D*) will contain $N/2$ coefficients. The DCT coefficients within a variable are independent in such a way that they can be coded as effectively as possible. This will further minimize the bit rate. The new variables are then split into two separate descriptions and sent over two different channels. The mentioned transformation is mathematically defined by Equation 4-9.

$$\begin{bmatrix} C \\ D \end{bmatrix} = T \begin{bmatrix} A \\ B \end{bmatrix} \text{, where } T = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ is the transformation matrix.}$$

Equation 4-9: MDPCT transformation

The transformation T controls the correlation between *C* and *D* which in turn control the redundancy in the MD coder. By performing the matrix multiplication in Equation 4-9, the new variables, *C* and *D*, can be calculated.

$$\Rightarrow \begin{cases} C = A + B \\ D = A - B \end{cases}$$

Equation 4-10: Variable C and D calculation

Equation 4-10 denotes a set of equations with two unknown variables, which can be calculated in the following way:

$$\begin{array}{l|rr} A + B = C & 1 & 1 \\ A - B = D & 1 & -1 \\ \hline \end{array}$$

$$\Rightarrow 2A = C + D \Rightarrow \underline{A = \frac{C + D}{2}}$$

Equation 4-11: Variable A calculation

$$\Rightarrow 2B = C - D \Rightarrow \underline{B = \frac{C - D}{2}}$$

Equation 4-12: Variable B calculation

Now, we have two expressions which may be used to reconstruct the original data perfectly by recovering *A* and *B*. A huge disadvantage with these two equations is that they require that both descriptions are perfectly received. Thus, if packet loss occurs during transmission, Equation 4-11 and Equation 4-12 are useless for reconstruction. Therefore, it is necessary to derive some new expressions which can be used for reconstruction of the original information. Since the transformation introduces statistically correlation between the two original variables, A and B (with calculated variances σ_A^2 and σ_B^2), it is possible to estimate A and B from either C or D. Based on [7], the following mathematically expressions for estimating lost information can be derived:

$$\hat{A}(C) = \frac{\sigma_A^2}{\sigma_A^2 + \sigma_B^2} \times C \qquad \hat{B}(C) = \frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2} \times C$$

Equation 4-13: Variable A and B estimation from variable C

$$\hat{A}(D) = \frac{\sigma_A^2}{\sigma_A^2 + \sigma_B^2} \times D \qquad \hat{B}(D) = -\frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2} \times D$$

Equation 4-14: Variable A and B estimation from variable D

In Equation 4-13 and 4-14, the variances σ_A^2 and σ_B^2 are given by:

$$\sigma_A^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(A_i - \overline{A}\right)^2 \text{ and } \sigma_B^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(B_i - \overline{B}\right)^2$$

Equation 4-15: Variances of variable A and B

In Equation 4-15, the mean values $\overline{A}$ and $\overline{B}$ are given by:

$$\overline{A} = \frac{1}{N}\sum_{i=1}^{N} A_i \text{ and } \overline{B} = \frac{1}{N}\sum_{i=1}^{N} B_i$$

Equation 4-16: Mean values of variable A and B

If one description (variable) is lost during transmission, the original variables, A and B, can be estimated with certain accuracy from the received variable, C or D, by means of the variances of A and B. This operation is given by Equation 4-13 and 4-14. If variable D is lost during transmission, Equation 4-13 gives the estimates of A and B from C. Similarly, if C is lost, Equation 4-14 gives the estimates of A and B from D.

This MD coding algorithm gives very good results, provided it is used in a correct way and on proper data. The effectiveness of this algorithm depends on the data in the two original variables, A and B. The data must contain a certain degree of dissimilarity. Several variants of this system were attempted during the development. In the first approach, the data were split into two descriptions by letting description one consist of odd-indexed coefficients and the other one of even-indexed coefficients, correspondingly the SDDDC system. Then, the two descriptions contain pretty similar data values and, of course, the mean values and the variances of the two variables were also pretty equal. Variable *C* was, by use of Equation 4-10, calculated by adding each value in A by the corresponding value in *B* (values with equal array-index). Similarly, except for the use of minus instead of plus, *D* was calculated. These calculations resulted in very high coefficient values in *C* and very low coefficient values in *D*. By loss of one description, Equation 4-13 and Equation 4-14 produced some extremely poor estimates of the original variables. This problem was diminished by having variable A consist of the larger DCT coefficients, and the variable *B* consist of the smaller coefficients. This is the reason why the Zigzag procedure was performed before the MD coding. It made sure that the DCT coefficients were ordered in a proper way before split into the two variables. In this case, variable *A* included the 32 first coefficients from each zigzag coded block and variable *B* included the 32 last bits from each zigzag coded block. Thus, variable *A* contained coefficients of high values and variable *B* contained coefficients of lower values (closer to zero). Then, variable *C* and *D* were calculated in the same way as above and became much more equal than in the first case. Further, this resulted in that the estimated variables from one received description became very good. *A* final development was accomplished by calculating *C* and *D* by adding and subtracting, respectively, the first value in *A* by the last value in B, the second value in A by the last but one value in *B* and so on. Unfortunately, this technique did not result in any better estimates of the original values and therefore it was abandoned.

Multiple Description Scalar Quantization

The last two MD coding systems developed are based on the principle illustrated in Figure 4-11.

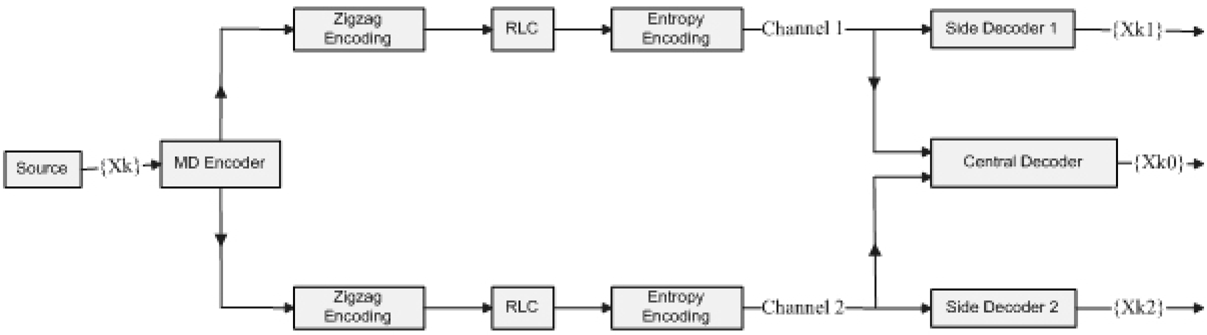


Figure 4-11: MDSQ overview

The difference from the MDPCT system is that the Zigzag encoding is performed in subsequence to the MD coding. Therefore, the source is here the output matrix of the DCT procedure and the quantization is performed within the MD encoder.

This system aims at using two separate scalar quantizers to produce two different descriptions, and an additional central decoder which makes use of both descriptions. A MDSQ encoder consists of a regular quantizer and an index assignment procedure. The MDSQ encoder produces, from each DCT sample, a pair of indices; i_1 and i_2 as illustrated in Figure 4-12.

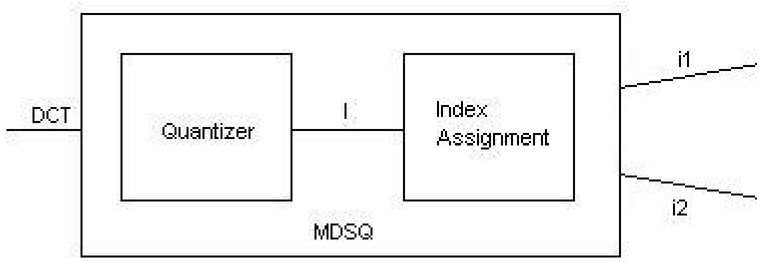


Figure 4-12: MDSQ encoder

The quantizer produces, from each DCT sample, a quantized value, $l, R \rightarrow l$. The Index Assignment takes the value l and produces the pair of indices (i_1, i_2). This is done by partitioning R into cells, and giving each cell an index. The partitioning and indexing are different for the assignment of i_1 and i_2. The indices are then individually Zigzag, RLC and entropy coded and sent over two different channels. The MDSQ can be broken up into two side encoders; $f_1 : l \rightarrow i_1$ and $f_2 : l \rightarrow i_2$, which selects the indices i_1 and i_2, respectively, and three decoders; $g_0 : (i_1, i_2) \rightarrow l$, $g_1 : i_1 \rightarrow l$ and $g_2 : i_2 \rightarrow l$. The central decoder

g_0 gives a finer estimate of l than g_1 and g_2. The estimated l is then dequantized.

The index assignment is an important part of the MDSQ and the optimization of the index assignment is very difficult [1]. In [3] Vaishampayan gave several heuristic techniques that give close to optimal performance. The basic idea is to number from upper-left to lower-right and to fill from the main diagonal outwards (see Table 4-11).

Table 4-11: Nested index assignment matrix

i_1

i_2

	-3	**-2**	**-1**	**0**	**1**	**2**	**3**
-3	0	1					
-2		2	3				
-1			4	5			
0				6	7		
1					8	9	
2						10	11
3							12

Nested Index Assignment

The first implementation was a scalar quantizer with nested thresholds (see Figure 4-13). Each quantizer outputs an index that can be used to make a rough estimate of the input DCT value.

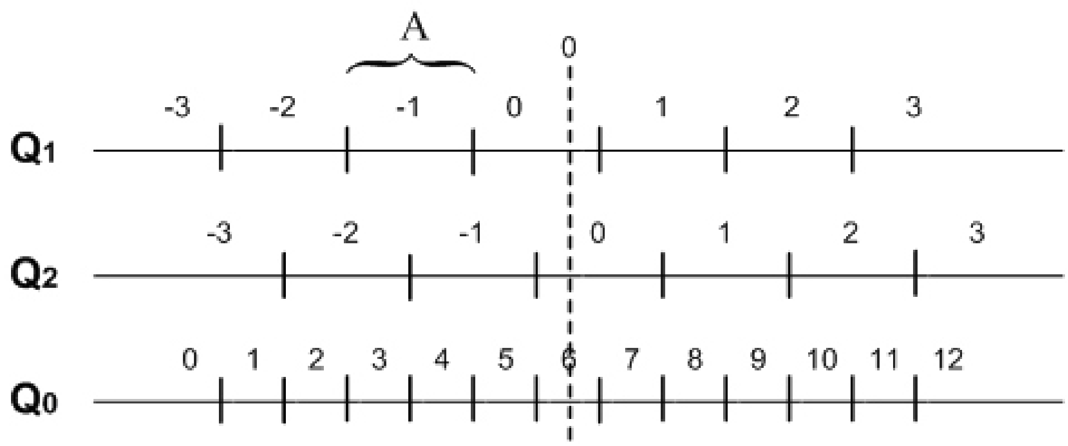


Figure 4-13: Nested index assignment quantizers

Example from Figure 4-13: $Q_1 : l \to \{-3,-2,-1,0,1,2,3\}$ and $Q_2 : l \to \{-3,-2,-1,0,1,2,3\}$. For a DCT sample input x; $f_1(x) = k_1$ and $f_2(x) = k_2$. The centroid of the cell is used for reconstruction. The central decoder uses both k_1 and k_2, and thus reconstructs to the centroid of the intersection cell $g_1(k_1) \cap g_2(k_2)$. The index assignment matrix for this example is given in Table 4-11, where i_1 identifies the column number and i_2 identifies the row number. The size of the cells, *A*, determines how accurate the reconstruction of l is. Larger cells give less accurate reconstruction, but also yield larger quantization, and thus reduce the bpp[1]. In this implementation the cell size of the side encoders and decoders is set to 2. This gives a cell size of 1 in the central decoder (there is no central encoder). Receiving both descriptions should now give the same quality as the baseline system, but with lower compression rate due to the redundancy in the descriptions. Since the RLC and entropy codes from the JPEG standard are used, setting the index number for each cell is not random. Since many values of l will be close to zero, the cell which includes 0 should have index number 0. The 0 cell is in the range $[-1.5,0.5\rangle$ for Q_1 and $[-0.5,1.5\rangle$ for Q_2. The cells on the left side of 0 are given increasing negative values, and the cells on the right side of 0 are given increasing positive values. This is in accordance with the JPEG RLC and entropy encoder. If only one description is received, 0, and not the centroid of the cell, is given as reconstructed l for index 0. This is to prevent a small positive l input in Q_1 to be reconstructed as a small negative l, and the other way around for Q_2. After dequantization this error could become huge.

Since the DC component is always the largest DCT coefficient, it determines the number of cells needed to cover all possible values of *l*. The DC component is at its peak when all components in a block are at 255 in a 24 bit colored image, or 8 bit gray scale image. From the DCT formula it can be shown that the maximum DC value is:

$$\frac{1}{4}\times\left(\frac{1}{\sqrt{2}}\right)\times\left(\frac{1}{\sqrt{2}}\right)\times(255\times 64)=2040$$

Equation 4-17: Maximum DC value

The maximum value of *l* is then 2040 when zero quantization is applied. The range of the indices must be large enough to cover this value. With a cell size of 2, this means that the number of cells need the capture all values in range [-2040, 2040] is 2040.

[1] Bits Per Pixel

Modified Nested Index Assignment

In the second implementation, the modified nested index assignment [3] was used (see Table 4-12). Some adjustments were made due to the use of the JPEG standard RLC and entropy coding. In [3] Vaishampayan shows that this modified nested index assignment gives the optimal combined exponential decay rates for the central and side distortions.

Table 4-12: Modified nested index assignment matrix

i_1

i_2	-3	-2	-1	0	1	2	3
-3	0	1					
-2	2	3	5				
-1		4	6	7			
0				8	9		
1					10	11	
2					12	13	15
3						14	16

Table 4-12 has some modifications in the matrix for $i_2 = -1$ and 0 and $i_2 =$ 0 and 1 due to the use of JPEG RLC and entropy coding.

For a given number of available side indexes (levels), the central distortion is smaller, at the cost of higher side distortion, than for the first implementation (compare Figure 4-13 and Figure 4-14). With 7 side indexes we get 12 central cells with nested index assignment and 16 with modified nested index assignment. The side encoders consist of cells with size 1 and 2. This gives a central cell size of 1. The range of the side index numbers should also here be large enough to cover a value of *l* up to 2040. An interval of length 6 uses two different indices with modified nested index assignment, and three with nested index assignment. The more efficient use of index numbers implies that the number of cells needed to do this is approximately $2/3$ of the number of indices needed by the first implementation. Due to the modification around the zero value it is not exactly $2/3$. The more efficient use of the indices means that receiving both descriptions should give the same quality as the nested index assignment and the JPEG comparison system (see Section 4.3.4) but at a rate (bpp) somewhere in between the two.

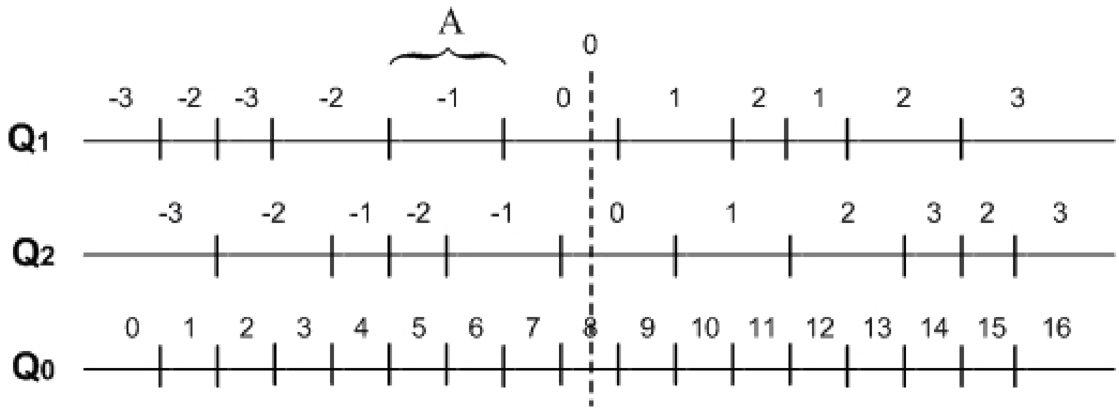


Figure 4-14: Modified nested index assignment quantizers

4.3.4 Comparison System

In order to measure the efficiency of the developed SD and MD coding algorithms it was also essential to develop a comparison system. Since the images, in addition to SD or MD coding, are exposed to lossy compression prior to transmission, it was unfavorable to measure the mentioned efficiency by comparing the original image with the reconstructed image. Since the purpose of this thesis was to evaluate MD coding as an image processing method, it was more interesting to measure the efficiency of the developed SD and MD coding algorithms than the efficiency of the complete systems. Therefore, a comparison system, that performs JPEG compression on a loaded image, was implemented to show the efficiency of the mentioned coding algorithms. Figure 4-15 illustrates the JPEG compression routine, based on [37].

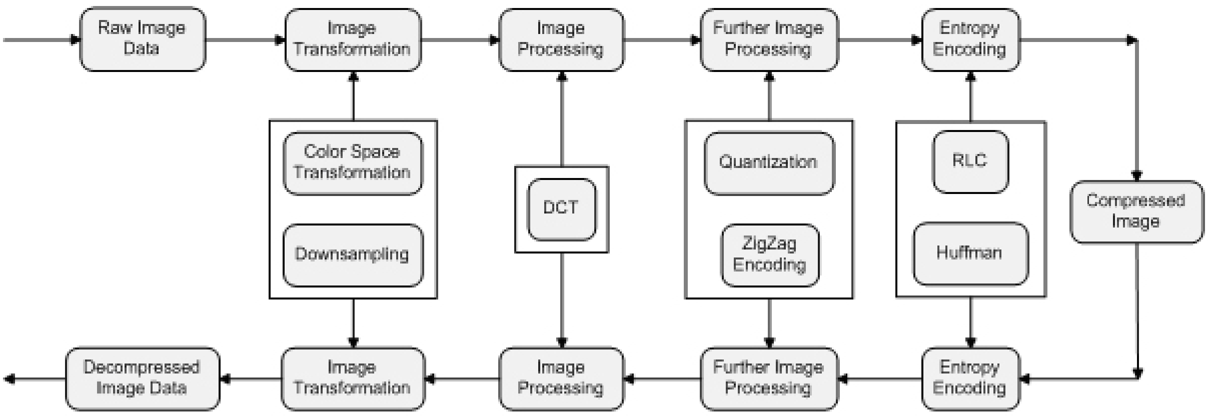


Figure 4-15: JPEG compression routine

This system performs only JPEG compression and decompression on a loaded image, corresponding to that performed in the above-mentioned SD and MD coding systems, and evidently no splitting or simulated network image transmission. From this, the results from the performed testing of all the developed systems were compared to the results from this JPEG routine instead of being compared to the original image.

4.4 Quality Measurement

4.4.1 Introduction

When using irreversible compression and transmission techniques which result in a degradation of visual quality in the reconstructed image, it is desirable to obtain the quality of the output image compared to the input image. In this section four different techniques for quality measurement of reconstructed (transmitted) images are described, both theoretically and mathematically. These techniques are MSE[1], PSNR, UQI[2] and MSSIM[3]. In this thesis, the PSNR and the MSSIM measures were finally chosen as the quantitative methods of comparing the image quality of the processed images with the original images. Basically, there are two different techniques for image quality measurement; subjective and objective measuring, which often are related to qualitative and quantitative measuring, respectively.

Subjective measuring is often performed by letting several human observers give a quality ranking of the achieved results. One disadvantage of this method is that each individual observer's opinion can be influenced by different factors, like environmental conditions, mood and motivation.

Objective measuring can be divided into two classes; Mathematical and HVS [38]. Several known methods for mathematical quality measurement exist, but the two most common are the earlier mentioned MSE and PSNR. The mathematical measuring methods possess several advantages. They are relatively easy to implement, they usually have low computational complexity, they are independent of viewing conditions and individual observers and they often result in a single quality value which gives a general idea of how good (or bad) the image quality really is. HVS considers Human Visual System characteristics to determine perceptual quality measures [39].

Subjective tests for assessment of image quality are often very resource demanding in practice because they usually involve a lot of peoples. Due to the lack of time, objective measuring methods were exclusively implemented in the test application, but some subjective opinions are also included in the discussion chapter.

[1] Mean Squared Error

[2] Universal Quality Index

[3] Mean Structural Similarity Index

4.4.2 Mean Squared Error and Peak Signal to Noise Ratio

The traditional measurement of image degradation is done by calculating pixel-to-pixel differences. The PSNR measuring method is strongly related to the above-mentioned MSE method [40] since they are defined as:

$$MSE = \frac{1}{N}\sum_{j=1}^{N}\left(I_j - \hat{I}_j\right)^2$$

Equation 4-18: Mean Squared Error

$$PSNR = 10\log_{10}\left(\frac{MAX^2}{MSE}\right) = 20\log_{10}\left(\frac{MAX}{\sqrt{MSE}}\right)$$

Equation 4-19: Peak Signal to Noise Ratio

In Equation 4-18, I denotes the original pixel values and $\hat{I}$ denotes the corresponding pixel values in the reconstructed (transmitted) image. By performing this subtraction for every pixel j, square the difference, and finally divide the sum of all the squares on the total number of image pixels, we get an average value for all the squared pixel differences in the whole image.

Dealing with 8-bit gray scale or 24-bit color images, the MAX value in Equation 4-19 equals 255. In this thesis, the PSNR was calculated on the basis of the red, green and blue values. By summing up the red, green and blue values before dividing the sums by three, an average value for all the three color layers was calculated [40]. As mentioned earlier, these techniques are performed by comparing the pixels in the original image with the corresponding pixels in the reconstructed image. This is supposed to give an estimate of how much damage has been done to the original image. The PSNR is measured in dB, which is a measure on a logarithm scale. Each 6 dB lost doubles the mean pixel-to-pixel difference between the two images. However, these mathematical measuring methods are not without shortcomings. It stands to reason that the viewing conditions play important roles in human perception of image quality. Therefore, these mathematical methods are slightly unreliable because they often result in poor correlation with the human perception of visual quality. This means that the measured result not necessarily agrees with the visual result. It rather gives a measure of the number of pixels which have been changed during the transmission.

4.4.3 A Universal Image Quality Index

In order to give an even better measure on the quality of the transmitted images, another technique called *A Universal Image Quality Index* (from now on called Q or Q-index) was implemented. "Universal" means that the quality measurement not necessarily depends on the image which is being tested, viewing conditions or individual observers. This is also a mathematical technique, but former research works [38][39] prove that this technique has some big advantages compared to MSE and PSNR, regarding the HVS. Two important advantages are that the Q-index outperforms MSE/PSNR for different types of image distortions and that it measures structure distortion which occurs during image degradation, rather than measuring the energy of errors.

Mathematical description of the Q-index

Assume that $x = \{x_i \mid i = 1,2,\ldots\ldots,N\}$ and $y = \{y_i \mid i = 1,2,\ldots\ldots,N\}$ are the original and the reconstructed image signals, respectively. Then, the Q-index is given by [38]:

$$Q = \frac{4\sigma_{xy}\mu_x\mu_y}{\left(\sigma_x^2 + \sigma_y^2\right)\left[(\mu_x)^2 + (\mu_y)^2\right]}$$

Equation 4-20: Q index definition

Detailed description of Equation 4-20:

- Mean value:

μ_x and μ_y denotes the mean value (average) in both the original and the reconstructed image, respectively. μ_x and μ_y are given by:

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i \quad \text{and} \quad \mu_y = \frac{1}{N}\sum_{i=1}^{N} y_i$$

Equation 4-21: Mean values of image signals X and Y

- Variance:

The variances σ_x^2 and σ_y^2 denotes the variances in picture x and y, respectively. A variance is a measure of the spread of the data values around the mean value. σ_x^2 and σ_y^2 are given by:

$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2 \text{ and } \sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \mu_y)^2$$

Equation 4-22: Variances of image signals X and Y

- Standard deviation:

σ_{xy} denotes the common standard deviation for both pictures. This is found by taking the square root of the multiplication of σ_x^2 and σ_y^2. σ_{xy} is given by:

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)$$

Equation 4-23: Standard deviation of image signals X and Y

The range of Q is [-1, 1]. The best value, 1, is achieved when the image data for x and y are identical, accordingly when $x_i = y_i$ for all $i = 1,2,....,N$. The lowest value, -1, occurs when $y_i = 2\mu_x - x_i$ for all $i = 1,2,....,N$.

To understand what the Q-index actually computes, the definition of Q can be rewritten as a product of three components.

$$Q = \underset{(a)}{\frac{\sigma_{xy}}{\sigma_x \sigma_y}} \times \underset{(b)}{\frac{2\mu_x \mu_y}{\mu_x^2 + \mu_y^2}} \times \underset{(c)}{\frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2}}$$

Equation 4-24: Rewritten Q index definition

The first component (a) is the linear correlation coefficient between x and y. The second component (b) measures how close the mean values are between x and y. The third component (c) measures how similar the variances of the signals are. Further description of the Q-index can be found in [38][39].

Q index calculation

The Q index is applied to images using a sliding window approach [39], with a window size of 8×8 pixels.

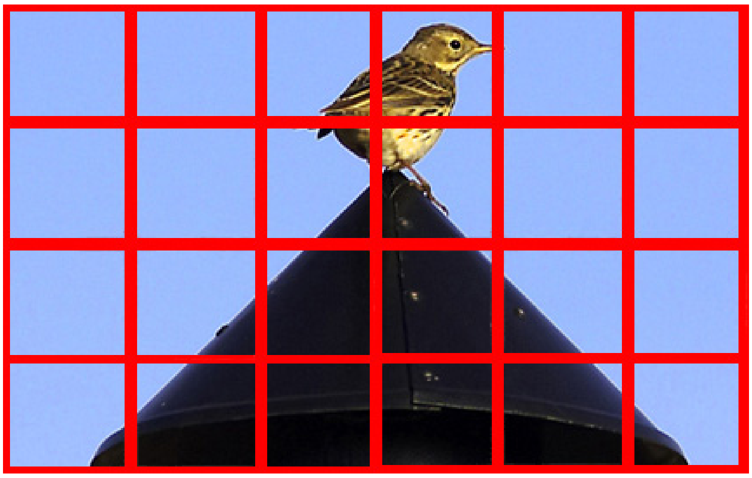


Figure 4-16: Sliding window approach

The sliding window starts in the upper-left corner and slides pixel by pixel horizontally and vertically until the lower-right corner is reached. For each step, the quality index is calculated within the sliding window. The overall quality index value is the average of all the calculated quality indices.

For instance, if the total numbers of steps are M, then the overall quality index value is given by:

$$Q = \frac{1}{M} \sum_{j=1}^{M} Q_j$$

Equation 4-25: Q index calculation using sliding window approach

In Equation 4-25, Q_j denotes the Q-index within the sliding window at the step j.

4.4.4 Mean Structural Similarity Index

During the project, a huge shortage in the Q-index technique was discovered. By some simulated image transmissions the Q-index became either way too low or inaccessible, which made the algorithm unstable. The reason to this was that images with nearly flat regions (bigger than the block size) resulted in values close to zero or equal zero in the variances (Equation 4-22) and the standard deviation (Equation 4-23). If both variances are zero, Equation 4-20 shows that this will give a denominator equal to zero, which makes the Q inaccessible. To avoid these problems, another technique was implemented. The new technique is a development of the Q-index technique, called MSSIM. Here, a short description of this development is given while [41] gives a detailed description of the technique

The MSSIM technique introduces two small constants, C_1 and C_2. According to [41], the constant C_1 and C_2 are introduced to avoid instability when $\left(\mu_x^2 + \mu_y^2\right)$ and $\left(\sigma_x^2 + \sigma_y^2\right)$ are zero or very close to zero. By means of these constants, MSSIM avoids the above-mentioned problems. C_1 and C_2 are given by:

$$C_1 = (K_1 L)^2$$

Equation 4-26: Constant C_1 calculation

$$C_2 = (K_2 L)^2$$

Equation 4-27: Constant C_2 calculation

In Equation 4-26 and 4-27, L denotes the dynamic range of the pixel values (255 for 8-bit image data), $K_1 = 0.01$ and $K_2 = 0.03$.

The SSIM index is then given by:

$$SSIM(x,y) = \frac{(2\mu_x \mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Equation 4-28: Structural Similarity index calculation

The SSIM index above corresponds to the Q index in Equation 4-20 if $C_1 = C_2 = 0$.

Similarly Equation 4-25, Equation 4-29 is used to calculate the MSSIM in order to evaluate the overall image quality.

$$MSSIM(X,Y) = \frac{1}{M}\sum_{j=1}^{M} SSIM(x_j, y_j)$$

Equation 4-29: Mean Structural Similarity index calculation

5 Implementation

5.1 Introduction

This chapter will basically describe the test application, which is the product that lies as basis of this thesis. In order to test whether the developed MD coding systems had the expected efficiency, a test application, where the user can load an image, perform image compression and SD or MD coding and simulate an image network transmission, was developed. To compare the efficiency of the MD coding systems with the baseline systems, the application makes it possible to simulate packet loss during the transmission. Finally, the application attempts to reconstruct the original image from the received information and reports the quality of the reconstructed image, compared to the original image, by means of PSNR- and MSSIM-measurements. It also reports the compression rate on the reconstructed image.

This chapter will further give a short description about our chosen programming language, C#, and development tool, Microsoft Visual Studio .NET, followed by a detailed description of application design, GUI[1] and the functionality of the developed test application.

5.2 The C# language and the .NET framework

The programming language C# (pronounced "see sharp") is a relatively new OOP[2] language that emphasizes a component-based approach to software development. While component-based programming has existed, in one form or another, for several years now, the vision of what C# enables has taken us to the next level in software development. C# has been designed in conjunction with the .NET Framework and is an adjustment of C++ which uses the Microsoft .NET framework to facilitate GUI design and optimize performance.

The .NET (pronounced "dot net") architecture is Microsoft's implementation of the CLI[3], plus several packages to support user interfaces, data and XML[4], web services, and a base class library. The primary purpose of the CLI is to facilitate the creation and execution of distributed components and services. It accomplishes this by enabling

[1] Graphical User Interface

[2] Object Oriented Programming

[3] Common Language Infrastructure

[4] Extensible Markup Language

programs written in different languages to operate together, giving programs the capability to describe themselves, and providing the execution environment to support multiple platforms. Figure 5-1 shows the .NET architecture [42].

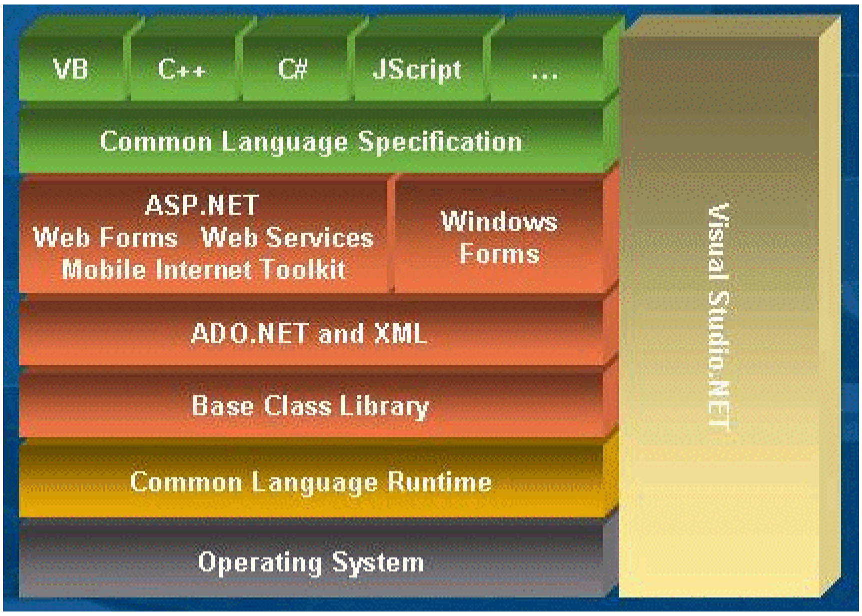


Figure 5-1: .NET architecture

The .NET framework software development kit ships with four programming languages: C++, Visual Basic, Jscript, and C#. The goal is to make them work interchangeably. While this is mostly true, there are differences between languages and the user must ultimately decide which language best suited his/her needs. Since the developers in this thesis are familiar with C# and Java (very similar to C#) from previous projects, and C# is superior to Java when it comes to designing graphical interfaces, the C# was considered to be the best suited programming language for this thesis.

5.3 Test application

The developed test application consists of a GUI where the different methods and parameters can be set, and an implementation of the different MD and SD coding systems. For simplicity both the encoding and decoding process is done by the same application object.

The encoder reads an image into memory, performs compression and SD or MD coding on the input image, and outputs two byte arrays which can be converted into packets suitable for network transmission. The test application does not include any network functionalities and, therefore, it is impossible to transmit the byte arrays over a physical network. These functionalities are not strictly necessary to evaluate MD coding as an image processing method for transferring information in error-prone networks with low transmission rate, related to quality, bit rate and file size, and therefore they were omitted.

The decoder performs the reverse encoder operations on the image data which were successfully received from the simulated network transmission. The decoder also applies the same compression and SD or MD coding techniques as the encoder to reconstruct the original image.

5.3.1 Application Design

The following UML[1] diagram shows the C# classes with their attributes and methods developed for the test application.

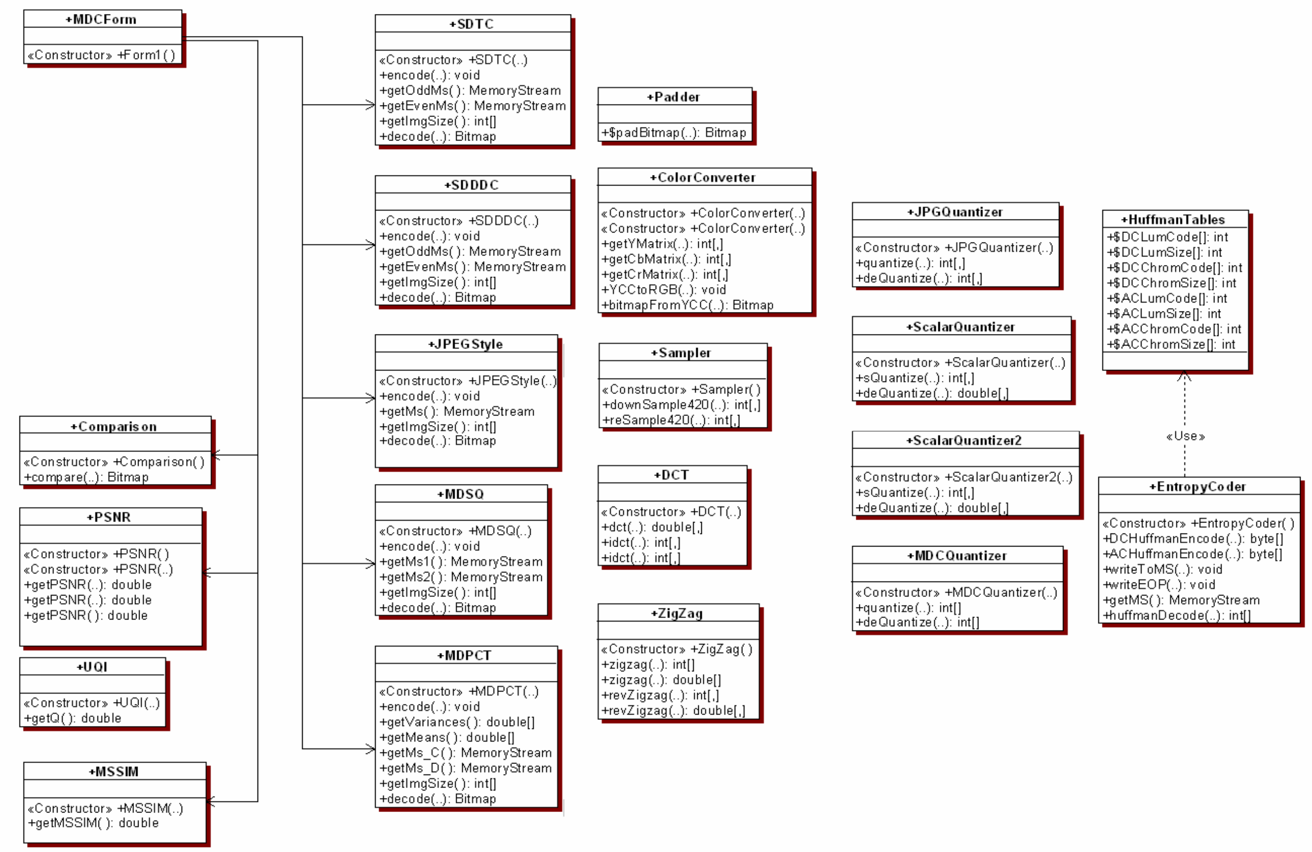


Figure 5-2: Class overview for the test application

The *MDCForm* class is the application GUI. It uses the desired coding class in the second column, indicated by the arrows, for encoding. They, in turn, run through all the compression steps in the third column. For the quantization, the different coding classes use different quantization classes. The comparison system, JPEG, and the baseline systems, SDTC and SDDDC, use the *JPEGStyle* class, the MDSQ and MDSQ_MOD use the *ScalarQuantizer* and *ScalarQuatizer2* classes, respectively, and MDPCT uses the *MDCQuantizer* class. Finally the two streams are entropy encoded by one *EntropyCoder* class each, except for the comparison system which codes only one stream. The decoding process runs through these steps backwards. When the image is decoded, its quality is measured by the *PSNR* and *MSSIM* class and the *Comparison* class constructs an image displaying the difference between the original and reconstructed image.

[1] Unified Modeling Language

5.3.2 Graphical User Interface and Functionality

The GUI shown in Figure 5-3 was developed in order to integrate the simulation and testing processes.

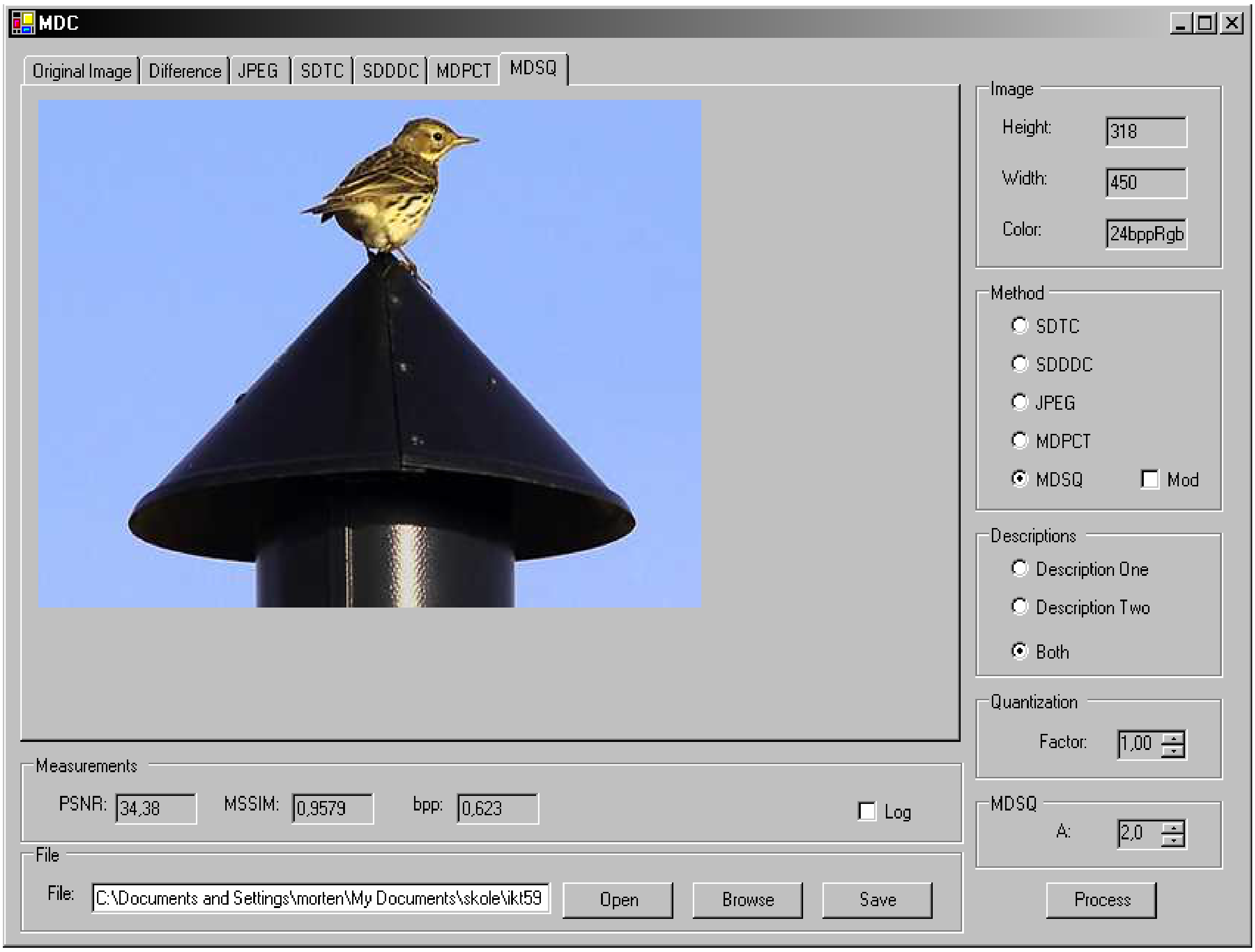


Figure 5-3: Test application GUI

In order to load an image into the image storage space in the middle of the window, the user may either specify the image location in the *File* textbox at the bottom and then click the *Open* button, or just click the *Browse* button. By clicking the *Browse* button a standard Windows "open file dialog" box appears and the user may locate the desirable image. The image is then displayed under the original image tab. Both the JPEG compressed image and the reconstructed images can be saved by clicking the *Save* button. Then a standard Windows "save file dialog" box appears and the user may choose a desirable location to save the image.

Once the image is loaded, the image height, width and color type are displayed in the *Image* box in the upper-right corner.

A simulated network image transmission is performed by clicking the *Process* button, after selecting a desirable SD or MD coding algorithm from the *Method* box. By clicking the *Process* button all the steps described in Section 3.2 (Figure 3-1) are performed with the selected SD or MD coding algorithm. See Appendix B for a technical description of the encoding and decoding process. In order to simulate packet loss during image transmission the user specifies whether to receive description one or description two in the *Description* box, prior to processing. If the user selects the *Both* button, both descriptions are successfully transmitted and no data are lost.

The GUI also allows the user to choose the degree of compression (quantization) by changing the *Factor* variable in the *Quantization* box. This factor is multiplied with the standard JPEG quantization table values. A higher factor means higher compression (quantization). The default *Factor* is set to 1.0.

When choosing the MDSQ from the *Method* box, the user can also change the size of the Scalar Quantizer cells by increasing the factor A in the *MDSQ* box. A larger A means larger cells, which in turn means less accurate reconstruction. The default A is set to 2.0. Increasing the A factor has the same effect as increasing the quantization factor. In fact, if you increase/decrease the quantization factor by a percentage, you get the same quality as if you increase/decrease A by the same percentage.

After the simulation is performed, the quality measurements, PSNR and MSSIM, and the bpp value of the reconstructed image are displayed in the *Measurements* box.

The user can also log the results of the simulations to a file, one text file for each method. The log file includes the PSNR, MSSIM, bpp, quantization factor, number of descriptions received and the filename. For MDSQ it also includes the A. Logs are saved in C:\log\[method]Data.txt. The directory is created if necessary.

The set of tabs above the image storage space are used to select between the original image, the reconstructed image for each chosen SD or MD coding method after processing and the difference between the last reconstructed image and the original image.

6 Results

6.1 Introduction

This chapter describes the test procedure in this thesis and presents the results of the tests. The results from the MD coding systems are presented, compared to the results from the baseline and comparison systems and annotated. In the simulated network transmission the descriptions correspond to streams of data packets in a conventional transmission system. There are not implemented any error detection or error correction routines in the test application and, therefore, it is assumed that each of the two descriptions is received without error or not received at all. In the application GUI, the user is able to specify which description he/she wants to receive; D1 [1], D2 [2] or both descriptions (D1 + D2). Finally, the original image is reconstructed from the successfully received description(s).

The qualities of the reconstructed images are measured by use of the objective mathematical techniques, MSSIM and PSNR. Subjective opinions are also taken into account in the discussion of the achieved results.

The remaining sections of this chapter describe the test procedure used in this thesis, a basis of the choice of image quality measurement technique, the results from all the tests and, finally, both a theoretical and a visual summary of the achieved results.

6.2 Test Procedure

In order to test the effectiveness of the different SD and MD coding algorithms, the following test procedure was carried out.

Three test images, with different level of detail, were used during the testing. Figure 6-1 displays the three test images.

[1] Description One
[2] Description Two


Bird.bmp


House.bmp


Dollarduck.bmp

Figure 6-1: Test images

The first image, *Bird*, is an image with a low detail level but it includes several straight lines and sharp contours. The second image, *House,* is an image with a slightly higher detail level and it also includes several sharp edges. The third image, *Dollarduck*, is an image with a high detail level. Especially the face of a watch and the banknote are elements that require a low degree of degradation in the quality of a reconstructed image to be displayed with an acceptable quality. The three test images are all 24-bit color, bitmapped images, and the varying detail level forms an idea about the typical properties of each coding technique. The tests in this thesis are performed on bitmapped images to ensure that the input image data is unaltered by the effects of compression, prior to image transmission simulation.

During the testing, all the test parameters and results were logged and the development tool, Matlab version 7.0.1, was used to plot the results. The results from both the baseline (SD coding) and MD coding systems are summarized over 2 pages. For each system, page one contains a short introduction and a figure illustrating the system performance for the three test images with corresponding comments. Here, system performance means the quality and the compression rate of the images, reconstructed from one and both descriptions. The bpp values from the different systems are presented to show whether the systems have unacceptably poor compression. One the second page, the three test images, processed by this particular coding technique and reconstructed from one and both descriptions, are displayed. In the testing of each individual coding technique, quantization factor 1 was chosen in order to compare the different techniques. However, all the techniques were also applied on the test images with different quantization factors. This is described and illustrated in Section 6.5. The comparison of all coding techniques are also performed, annotated and displayed in the summary section of this chapter.

6.3 PSNR versus MSSIM

When doing the tests, the three quality measurement techniques mentioned earlier, PSNR, UQI and MSSIM, were logged. Due to the lack of stability and relation to HVS in UQI and PSNR, respectively, the measured MSSIM values were plotted in Matlab for visualization and planned used as a basis for the results and discussion of the tests.

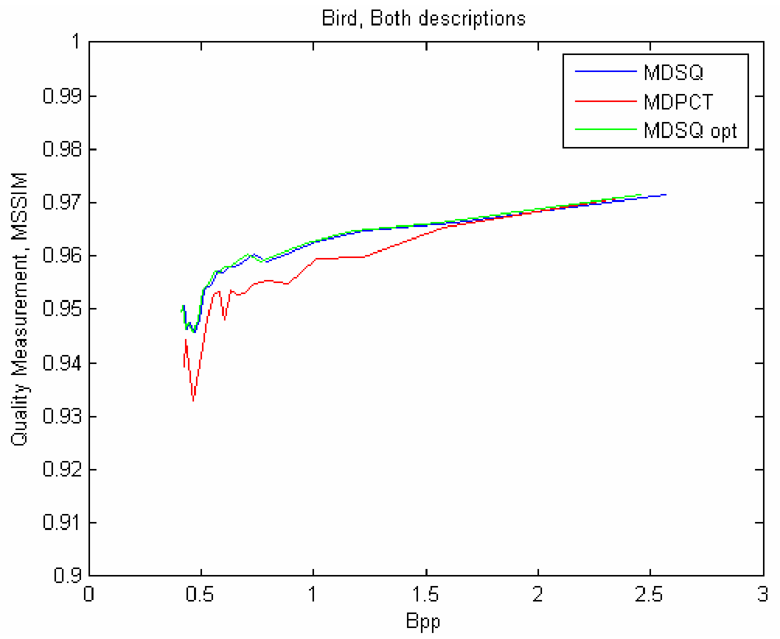


Figure 6-2: MSSIM receiving both descriptions from Bird

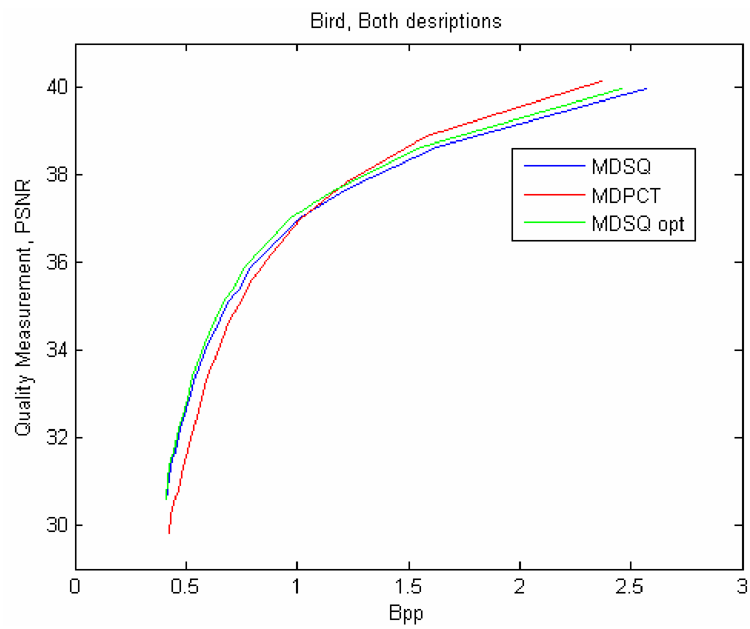


Figure 6-3: PSNR receiving both descriptions from Bird

When plotting the MSSIM values calculated from the reception of both descriptions from *Bird*, some unevenness in the graphs were detected. As seen in Figure 6-2, at some points, a slightly higher compression, gives better quality according to MSSIM. Since larger compression gives less accurate reconstruction of a value, and hence a reconstructed image of lesser quality, it seems that the MSSIM still has some problems with large flat regions. The constants introduced in MSSIM deals with the problem of divide by zero when both variances are zero (see Equation 4-28), but does not completely fix the problems with low variances. In Figure 6-3, the MSSIM values have been replaced with the PSNR values. Here we see fine smooth curves with quality always degrading with larger compression. Based upon the problems with both the UQI and the MSSIM, the final choice for quality measurement technique was the PSNR.

6.4 Results of image processing with different coding techniques

6.4.1 Baseline Systems

As mentioned earlier, the baseline (SD coding) systems were developed to prove the need for retransmissions in conventional image transmission systems when packet loss occurs. This need is clearly illustrated on the following pages.

Single Description Transform Coding

The SDTC system, described in Section 4.3.2, is the simplest system developed in this thesis. The system has no possibility of estimation of lost data packets and it is therefore extremely vulnerable against packet loss. Figure 6-4 illustrates an overview of the quality achieved by transmission of the three test images (upper subplot) and the corresponding compression for each image (lower subplot).

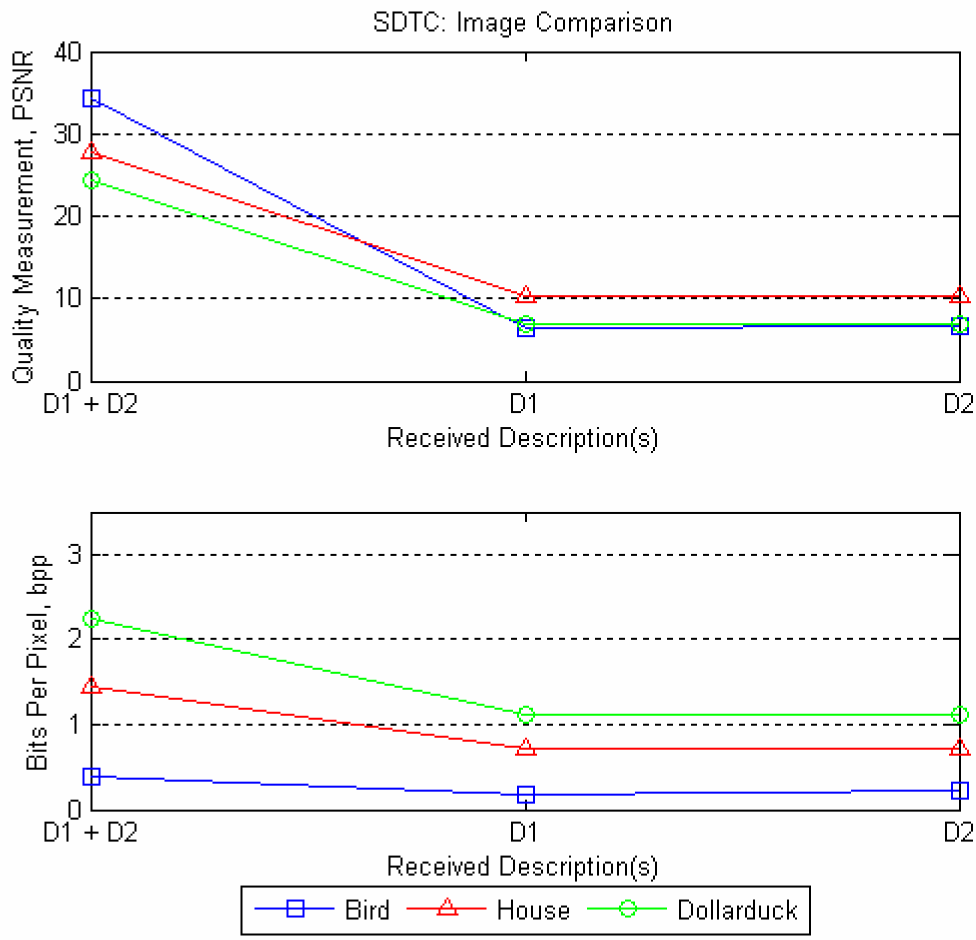


Figure 6-4: SDTC - System performance

As the figure above illustrates, this system can provide a very useful reconstruction of the original, transmitted, image if both descriptions are successfully received. However, a sharp drop in quality occurs as soon as one description is lost. This quality reduction occurs on all the test images and is caused by the systems' inability to estimate the lost data. The compression rate in this system is good since it does not generate any redundancy.

Figure 6-5 illustrates the SDTC system performance on the three test images. The left column shows the reconstructed images when no descriptions are lost, while the right column shows the reconstructed images by loss of one description, 50 percent packet loss. The lost data are replaced by black blocks.

Figure 6-5: Visual results from SDTC

Single Description Double DC

The SDDDC system, also described in Section 4.3.2, is an improvement of the first system. This system has neither any function for estimation of lost data, but since a small degree of redundancy is introduced in the original image data, the quality of the reconstructed image, by 50 % packet loss, is much better than by use of the SDTC system. Therefore, this system is slightly more resistant to packet loss. Despite this improvement, the reader will soon observe that the quality of the reconstructed images is still unacceptably poor. Figure 6-6 illustrates the system performance for the three test images.

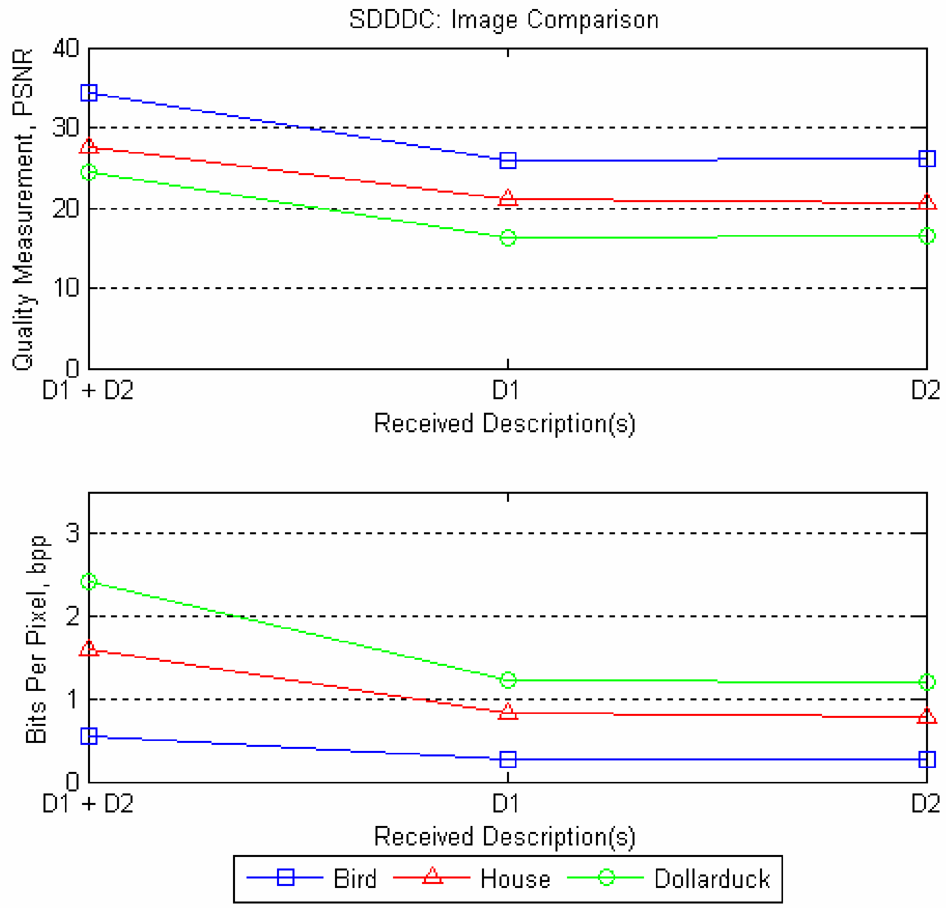


Figure 6-6: SDDDC - System performance

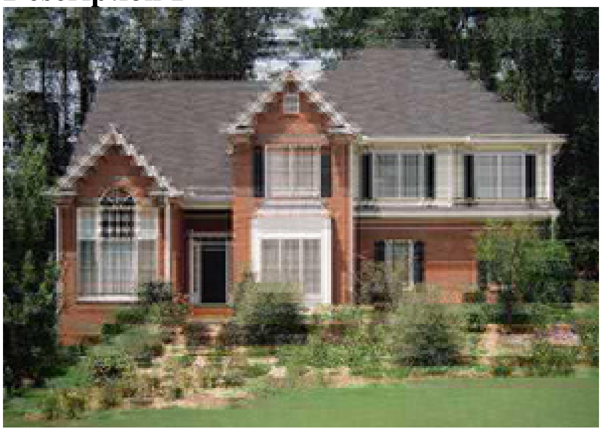
As the figure above illustrates, this system can also provide a very useful reconstruction of the original, transmitted, image if both descriptions are successfully received. The drop in quality of the three images, by loss of one description, is still unacceptably high. The compression rate is here slightly lower (higher bpp) for all the images because of the small amount of redundancy which is generated.

Figure 6-7 illustrates the SDDDC system performance on the three test images. The left column shows the reconstructed images when no descriptions are lost, while the right column shows the reconstructed images by loss of one description.

Description 1+2

Description 1

Description 1+2



Description 1

Description 1+2

Description 1

Figure 6-7: Visual results from SDDDC

6.4.2 MD Coding Systems

Multiple Description by Pairwise Correlating Transform

The MDPCT system, described in Section 4.3.3, is the first MD coding system developed in this thesis. The system is robust against packet loss since lost data are estimated in a satisfactory way. Figure 6.8 illustrates the MDPCT system performance for the three test images.

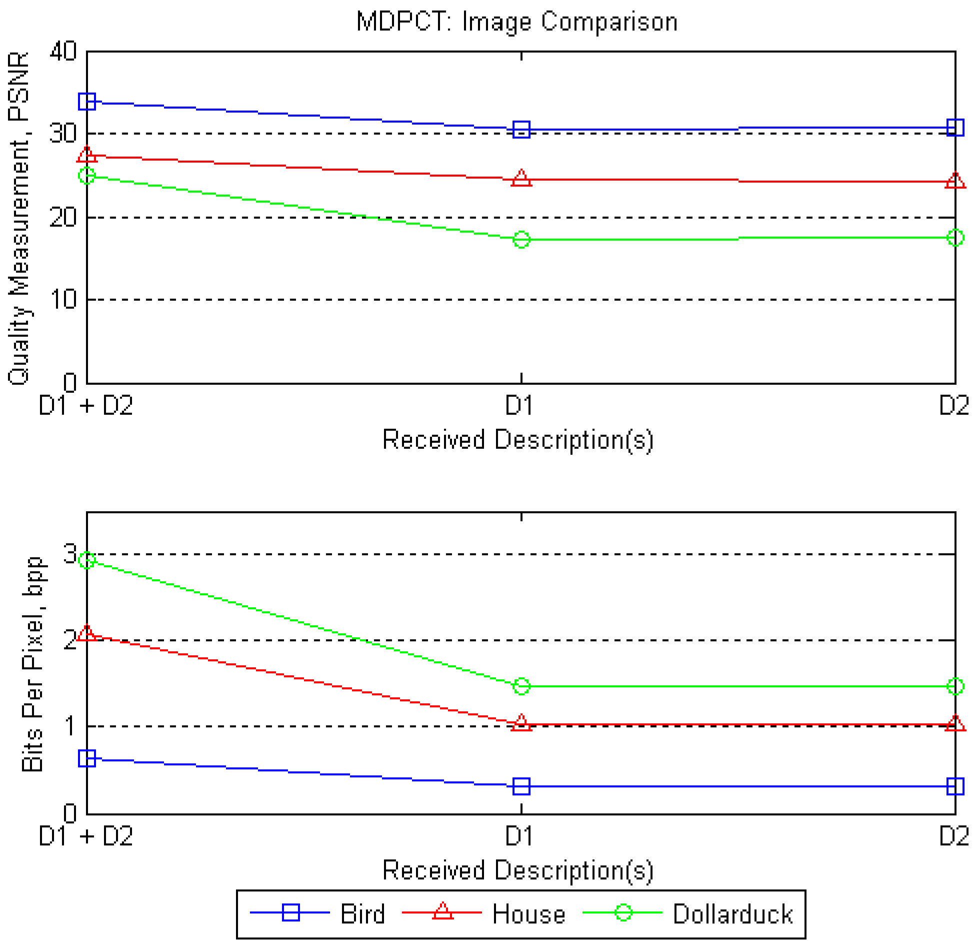


Figure 6-8: MDPCT - System performance

As the figure above illustrates, this system can, as opposed to the two baseline systems, provide a satisfactory reconstruction of the original image, even if one description is lost during transmission. In that case, the quality decreases with approximately 3 dB on the two less detailed images, shown in blue and red, and with approximately 7 dB on the most detailed image. Because of the increased amount of redundancy introduced in the original image data prior to transmission, the compression rate is here a bit lower than it were in the baseline systems.

Figure 6-9 illustrates the MDPCT system performance on the three test images. The left column shows the reconstructed images when no descriptions are lost, while the right column shows the reconstructed images by loss of one description.


Description 1+2 **Description 1**


Description 1+2 **Description 1**


Description 1+2 **Description 1**

Figure 6-9: Visual results from MDPCT

Multiple Description Scalar Quantization

The MDSQ system, described in Section 4.3.3, is the second developed MD coding system. This system produces an even better estimate of lost data than the MDPCT system does, despite that the compression rate is approximately the same. Figure 6-10 illustrates the MDSQ system performance for the three test images.

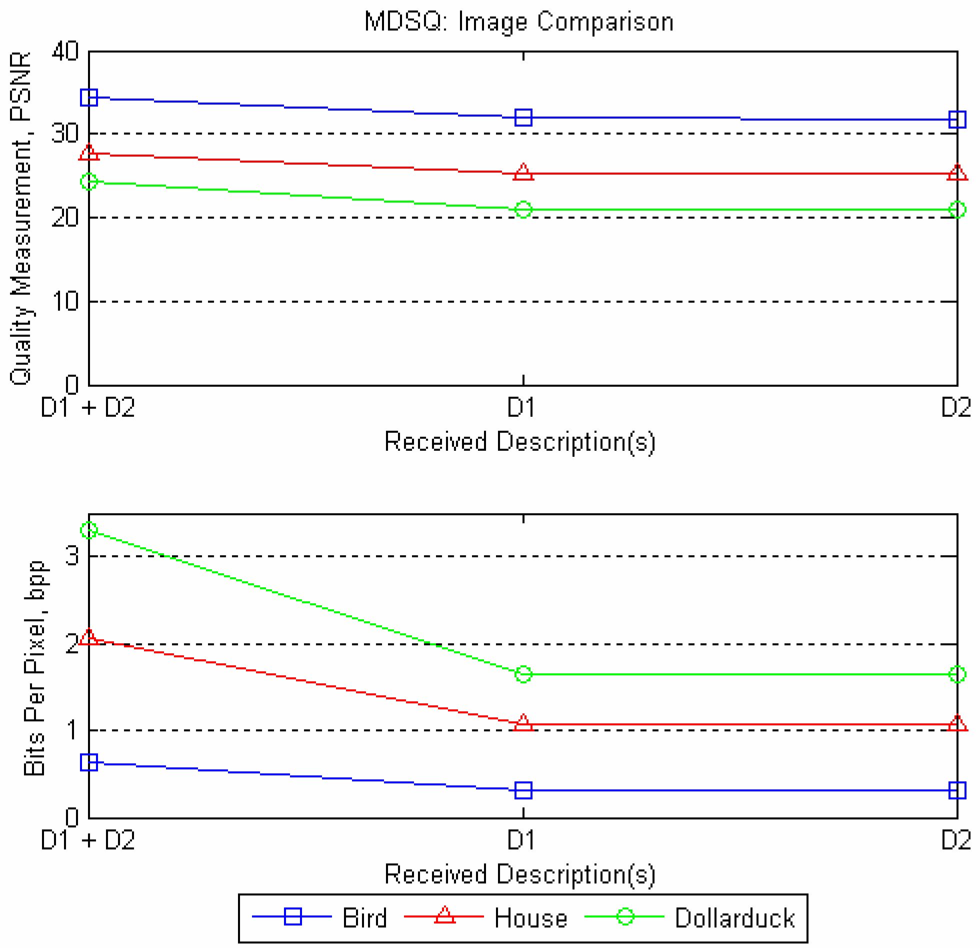


Figure 6-10: MDSQ - System performance

As the figure above illustrates, the quality of all the transmitted images is better than by the previous MD coding system. The compression rate on the low detailed image, shown in blue, is a little higher than by MDPCT while it is a little lower for the last two images.

Figure 6-11 illustrates the MDSQ system performance on the three test images. The left column shows the reconstructed images when no descriptions are lost, while the right column shows the reconstructed images by loss of one description

Description 1+2

Description 1

Description 1+2

Description 1

Description 1+2

Description 1

Figure 6-11: Visual results from MDSQ

Multiple Description Scalar Quantization_Modified

The MDSQ_MOD system, also described in Section 4.3.3, is the third, and last, developed MD coding system. The system is, as mentioned earlier, a modification of the MDSQ system. The objective was to increase the compression rate without decreasing the quality when both descriptions are received, and an acceptable decreasing of quality when one description is received compared to the MDSQ system. Figure 6-12 illustrates the MDSQ_MOD system performance for the three test images.

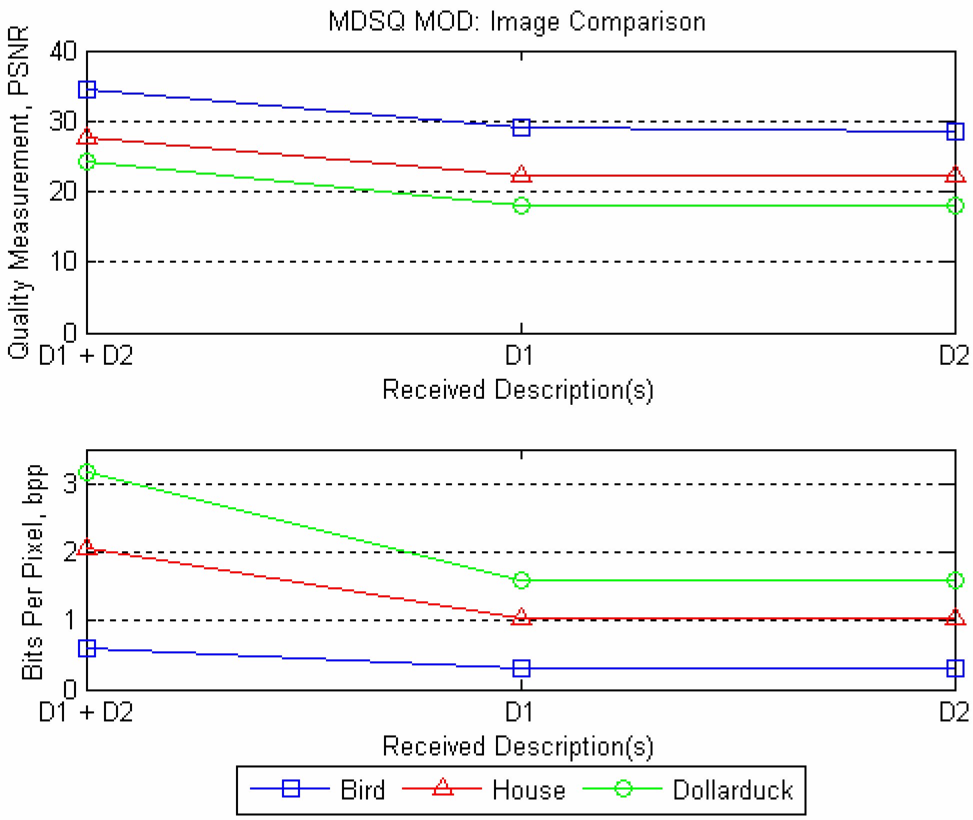


Figure 6-12: MDSQ_MOD - System performance

As the figure above illustrates, the quality of all the reconstructed images is a little poorer than by the previous MD coding system. The compression rate is, however, a bit higher but not high enough to consider this system as an improvement of the original MDSQ system.

Figure 6-13 illustrates the MDSQ_MOD system performance on the three test images. The left column shows the reconstructed images when no descriptions are lost, while the right column shows the reconstructed images by loss of one description

Description 1+2

Description 1

Description 1+2

Description 1

Description 1+2

Description 1

Figure 6-13: Visual results from MDSQ_MOD

6.5 Results of image processing at different compression rates

Figure 6-15 illustrates the compression and achieved quality on *House*, processed by the three MD coding systems, when D1 (upper subplot) or both descriptions (lower subplot) are received. The different compression rate was modeled by using different quantization factors (see Section 5.3.2) between 0.1 and 3.0. See Appendix D for equivalent figures for the last two test images.

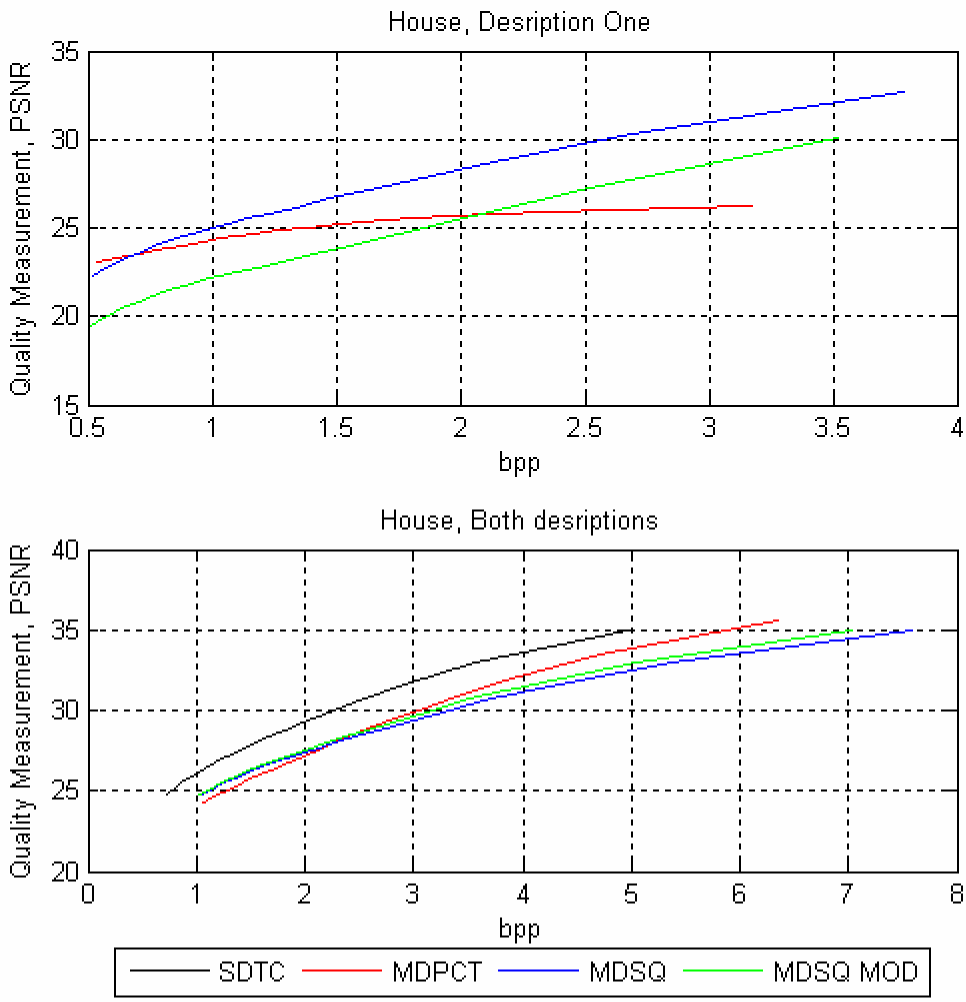


Figure 6-14: Compression and quality in the MD coding systems

As the upper subplot in Figure 6-15 illustrates, the quality of the reconstructed image nearly increases in inverse ratio with the compression rate, if one of the coding techniques, MDSQ or MDSQ_MOD, is used. On the other hand, if MDPCT is used, the quality is steadying close by 25 dB. When both descriptions are successfully transmitted (shown in the lower subplot), the quality measurements from all the MD coding techniques increase when the compression rate decreases. Unfortunately, it seems that the redundancy in the MD coded image is a little too high, compared to the SDTC baseline system (shown in black) which, in the both-description case, corresponds to the JPEG compressed image. Since all the SDTC values lay around 10 dB in the one-description case, they are omitted in the upper subplot.

6.6 Summary

6.6.1 Theoretical Summary

In order to compare the different coding systems, their performances are plotted in the same diagram, Figure 6-14, below. This figure illustrates the results, both within quality and compression, the different coding systems produce when the second test image, *House*, is processed. See Appendix C for equivalent figures for the last two test images.

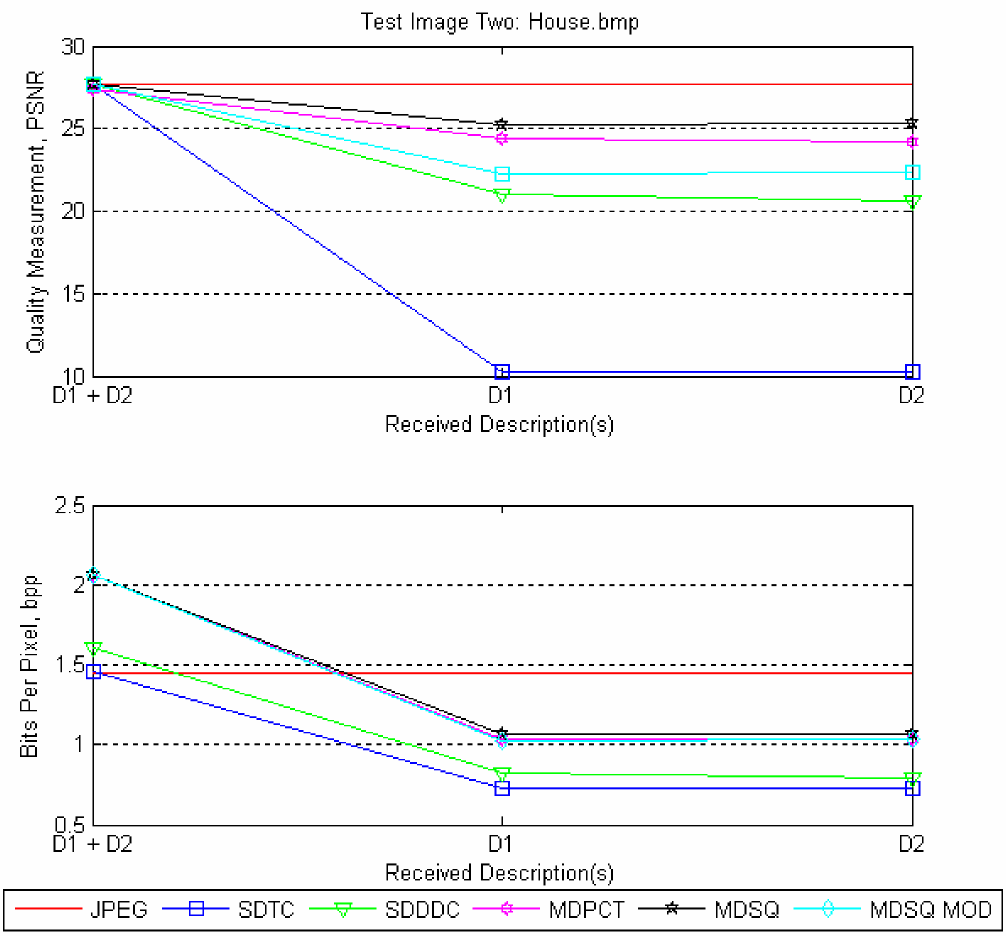


Figure 6-15: Comparison of all system performances

As the reader may observe, all the MD coding systems perform significantly better than the baseline systems. The red line in the upper and lower sub plot illustrates the quality and the compression of the JPEG compressed image, respectively. The quality of the compressed image gives the reader a picture of the quality degradation in consequence of the different coding techniques.

Figure 6-14 illustrates that the MDSQ, shown in black, provides the best quality by loss of one description. The other two MD coding systems, MDPCT and MDSQ_MOD, follow second and third, respectively, before the baseline systems, shown in blue and green, provide the poorest results, as expected. The lower subplot illustrates that the introduced redundancy in the MD coding systems involve some effects in the compression rate. The small amount of redundancy introduced in the SDDDC system is also clearly shown in the figure.

6.6.2 Visual Summary

The following pages contain the three test images processed by both the best baseline system, SDDDC, and the, apparently, best MD coding system, MDSQ. The images illustrate the reconstructed images from one received description.

Figure 6-16: SDDDC - Description one

Figure 6-17: MDSQ - Description one

Figure 6-18: SDDDC - Description one

Figure 6-19: MDSQ - Description one

Figure 6-20: SDDDC - Description one

Figure 6-21: MDSQ - Description one

7 Discussion

7.1 Introduction

This thesis started with some expectations and goals for the forthcoming results, and the following chapters gave a relatively detailed description of the necessary and employed theory about compression, SD and MD coding, and quality measurement. Further, this theory was implemented in an application used to test the different coding techniques on different test images in order to obtain useful results for further analysis and discussion.

The purpose of the work has been to show that MD coding may reduce the need for, and the problems related to, retransmission in error-prone transmission systems by a considerable degree. MD coding makes use of successfully transmitted data, without the need for retransmission of lost data. This technology is well suited for the transmission of data which is useful at many different levels of quality, e.g. voice, sound, image and video. Due to the time limit and the researchers' limited background knowledge on this subject, this thesis has focused only on the transmission of still image data. The work presented in this thesis serves as a basis and could be extended to other areas of application mentioned above.

In the previous chapter the results of the tests were presented. In this chapter the discussion of the achieved results is continued, and the results are also compared with the targets and goals proposed in Section 1.4.

The following issues will be discussed in the remaining parts of this chapter:

- **Results of image processing with different coding techniques**
- **Results of image processing at different compression rates**
- **Future work**

7.2 *Results of image processing with different coding techniques*

The table below substantiates the earlier described theories and gives a clear picture of the improvements and advantages MD coding involves, compared to conventional SD coding.

Table 7-1: Summary - System performances

System	Quality from one versus both descriptions			Redundancy		
	Bird	**House**	**Dollarduck**	**Bird**	**House**	**Dollarduck**
JPEG	-----	-----	-----	(0.388 bpp)	(1.447 bpp)	(2.227 bpp)
SDTC	81 % (28 dB)	63 % (17 dB)	72 % (17 dB)	2 %	0.5 %	0.4 %
SDDDC	25 % (8 dB)	24 % (7 dB)	33 % (8 dB)	40 %	11 %	9 %
MDPCT	11 % (3 dB)	11 % (3 dB)	30 % (8 dB)	63 %	43 %	32 %
MDSQ	7 % (2 dB)	10 % (2 dB)	14 % (3 dB)	61 %	42 %	48 %
MDSQ_MOD	19 % (5 dB)	22 % (5 dB)	34 % (6 dB)	56 %	42 %	42 %

7.2.1 Baseline Systems

As stated earlier, the first baseline system, SDTC, is extremely vulnerable to packet loss. Table 7-1 clearly shows this, since the system experiences a dramatically drop in quality by the loss of one description. Loss of one description resulted in a quality reduction of approximately 81 % (28 dB), 63 % (17 dB) and 72 % (17 dB), compared to the reception of two descriptions, for the three test images, *Bird*, *House* and *Dollarduck*, respectively. These achieved quality measures lay outside the specified targets from Section 1.4. However, this agrees with the visual results in Figure 6-5, where half the image data (blocks) is black. This enormous drop in quality occurs because the system is incapable of detecting and correcting errors, and does not introduce any form of redundancy in the original image data. Therefore, it is impossible to estimate lost or erroneous data from that successfully received. This will significantly affect the PSNR and the perceived quality of the reconstructed image.

The table also shows that the redundancy in the SDTC system, when compared with the JPEG system, is very small since no redundancy is actually introduced into the original image data. The splitting of the images results in a less effective compression, which leads to an insignificant decrease in the compression rate.

This system is evidently dependent on retransmissions in order to be effectively applied in error-prone networks. The most natural development is to introduce a certain amount of redundancy in such a way that retransmissions are avoided in spite of packet loss. This was the base for the next developed system that of SDDDC.

The SDDDC system is an improvement of the SDTC system. Since all the DC coefficients are sent in each of the two descriptions, a whole picture of the transmitted image is reconstructed, as opposed to the SDTC system where half the image became black. The achieved PSNR measures are also close to the theoretical targets, stated in Section 1.4. By loss of one description, a quality degradation of 25 % (8 dB), 24 % (7 dB) and 33 % (8 dB) occurs on the three test images, *Bird*, *House* and *Dollarduck*, respectively. These results do not agree very well with the visual results. As Figure 6-7 shows, the images, reconstructed from one description, are notably blurred and diffuse. This is especially clear on images with sharp edges, for example edges found in the *Bird* image.

Since all the DC coefficients are duplicated and sent in each description, the redundancy in this system is much higher than that in the SDTC system. For the less detailed image, the redundancy increases with 40 % (0.15 bpp) compared to the JPEG system. However, the redundancy on the two most detailed images are within the stated goals, with an increase of 11 % (0.16 bpp) and 9 % (0.2 bpp), respectively. However, these redundancy measures are relatively insignificant since the visual qualities on the reconstructed images are unacceptably poor.

Akin to the previous system, this system is also dependent on retransmissions in order to offer an acceptable quality on the reconstructed images. A natural development of this system is to introduce a certain degree of redundancy, in the form of correlations, between the two descriptions. It will then be possible to estimate a lost description from the received description without sending duplicated image data on both of the descriptions. This is the basis for MD coding, and also that of the three next systems, namely the MDPCT, MDSQ and MDSQ_MOD system.

7.2.2 MD Coding Systems

All the MD coding systems are considerable improvements of the baseline systems. The reduction in quality by loss of one description is, in most cases, much smaller than in the baseline systems. Unfortunately, the systems are not optimal, as described below. The MDPCT system produces very good results on the two less detailed images, but notably has some problems with the high detailed image. Table 7-1 shows that loss of one description result in a reduction in quality of 11 % (3 dB) on the first two images, which is within the stated targets from Section 1.4. On the third image, *Dollarduck*, a degradation of 30 % (8 dB) occurs, which is barely within the bound of the stated targets. This system gives good visual results of the reconstructed images. By equal quantization, the redundancy in this system is significantly higher than in the above-mentioned systems. Compared to the JPEG system, redundancies increase with 63 % (0.24 bpp), 43 % (0.62 bpp) and 32 % (0.71 bpp) for *Bird*, *House* and *Dollarduck*, respectively. Unfortunately, these values are much higher than the stated goals from Section 1.4.

The MDSQ system achieves very good image quality despite the loss of one description. The reduction in quality from 7 % (2 dB) to 14 % (3 dB) is within the stated target and the visual qualities of the reconstructed images (Figure 6-11) are also outstanding. This system also achieves a higher redundancy than that expected. The increases in bpp, from 42 % to 61 %, for the three test images are also much higher than the stated goal of 30 %. However, since this system achieves significantly higher quality on the reconstructed image than the MDPCT system does, the conclusion is that MDSQ is an evident improvement from the MDPCT

The last MD coding system did not become the predicted improvement of the MDSQ system. The redundancy was slightly reduced, but since the quality on the reconstructed images also decreased, this system can not be called an improved system. Nevertheless, all the MD coding systems achieve fractionally increased redundancies, where they were all considerable improvements of the baseline systems. Clearly, the MD coding systems make it possible to reconstruct the images, with an acceptable quality, by loss of one description (which again means 50 % packet loss). The high redundancies can simply be reduced by increasing or decreasing the quantization factor, but this will in turn affect the reconstruction quality. In Section 7.3 image processing with different degrees of compression (quantization) will be discussed.

7.3 Results of image processing at different compression rates

In this section the results of compressing the three different MD systems at different rates are discussed. This is reviewed by comparing the systems to each other, then to the baseline systems and to the goals set in Section 1.4. First, the reception of one description is discussed, and then subsequently the reception of both descriptions is evaluated.

The upper subplot in Figure 6-15 shows the compression and quality of the *House* image when one description is lost. This indicates that the MDPCT is stable around 25 dB for all compression rates. Only when compression is dramatically high will the image quality drop significantly. This is also the case for the other two test images, viz. *Bird* and *Dollarduck*, the only difference being that the stable levels lies higher for *Bird* and lower for the *Dollarduck*. This stability of quality, in spite of different compression rates, is caused by the inaccurate computation of some variable values. When an image contains a high level of detail, it could be the case that many coefficients from the variable *B* are actually higher than its pairing partner in variable *A* (see Section 4.3.3). The *A* variance will always be bigger than the *B* variance, thus when receiving a transformed variable (*C* or *D*), the computation may reconstruct *A* as e.g. 98 % of C (or D) and *B* as 2 %. This bias completely overshadows the quantization inaccuracy. The MDSQ accuracy lies steadily higher than the MDSQ_MOD when one description is lost. This is as expected, since MDSQ_MOD improves the quality-compression rate when both descriptions are received at the expense of a decrease in quality when one description is lost. The MDSQ and MDSQ_OPT are superior to the MDPCT at low compression rates, however at high compression rates MDPCT appears to perform better. This is the case for all the test images, but the reason for this is for the time being dubiously. It also seems that the more detailed an image is, the worse the MDPCT is, compared to the MDSQ and MDSQ_MOD.

Figure 6-15 shows the compression and quality of the *House* image when both descriptions are received. This plot indicates that the MDPCT is at its best at low compression rates, and that MDSQ_MOD is at its best at high compression rates. The MDSQ_MOD is only marginally better than MDSQ over all the compression rates, this difference is much higher in favor of MDSQ when receiving only one description. This indicates that MDSQ is more suitable when the probability of packet loss increases, compared to MDSQ_MOD. It seems that the modification of the MDSQ has not been successful and that, overall, MDSQ performs better than MDSQ_MOD. The MDPCT becomes significantly better the more detailed the image is, when compared to MDSQ and MDSQ_MOD. It can also be seen that with the *Dollarduck* test image the MDPCT performs better at all compression rates.

The JPEG system is plotted in Figure 6-15. By comparing this system to the different MD classifications, the redundancy can be found. Indeed, the redundancy increases with the compression rates, except for MDSQ that decreases with higher compression rate, but only on the test images *House* and *Dollarduck*. The redundancies using MDPCT vary from approximately 70 % on the *Bird,* to about 24 % on the *Dollarduck*; it appears that the redundancy increases with less detailed images. The MDSQ_MOD system has redundancies that vary from about 40 % to 64 %, while the MDSQ redundancies vary from about 46 % to 65 %.

The comparison of the qualities and redundancies of the three MD systems to the targets set in Section 1.4 clearly indicates that the redundancy is much too high for almost all the compression rates and all the systems. It is found that with the low detailed image (viz. *Bird*) the redundancy becomes especially high. Only low compression rates using MDPCT on the images *Dollarduck* and *House* fall within the redundancy target set out in Section 1.4. When also taking the quality targets into consideration, only the lowest compression rate, quantization factor 0.1 (see Section 5.3.2), with MDPCT on *House,* falls within the targets.

7.4 Future Work

7.4.1 Introduction

Due to the availability of time, it was not viable to investigate, develop or implement all the relevant areas of the assignment as set out in the beginning of this thesis. In this section several improvements and changes, as well as other issues which could be subject to further work, will be proposed and discussed.

7.4.2 General

The main purpose of this thesis is to provide strong evidence that MD coding is a suitable technique for image transmission in unreliable transmission systems. A very exciting area of future work is to employ MD coding to other multimedia applications, such as video streaming. As stated earlier, network congestion is already an important issue that affects the performance in image transmissions, but it will definitely become an even bigger problem in services that require even higher bandwidth, such as video streaming. As video recording is composed by many freeze-frames, and since small degradations on some frames are invisible to the human perception, MD coding ought to be suitable for such services.

7.4.3 Compression

Compression is another important part of this thesis and there are excellent possibilities for improvements within this matter. An investigation of other, and possibly better, compression techniques will be of great interest with a view to improve the efficiency of the overall system. According to Section 4.2.9, the most natural development will possibly be to implement wavelet compression (see Section 4.2.9), since previous research works [43][44] have proven that this is a better, although a more complex, compression technique than the DCT technique used in this thesis.

7.4.4 MD Coding

An integral part of further work on MD coding will be to develop and test other MD coding algorithms, in order to find the optimum algorithms for different applications. It will also be of great interest to employ algorithms that split a signal into more than two descriptions, this gives the user more options, such as specifying of the amount of data to be lost during image transmission, or which descriptions are received and lost. This will give a more realistic picture of reality than the simulations covered within this thesis, as it is impossible to simulate other quantities of packet loss, other than 50 % or no packet loss at all. The reason for this is, as earlier mentioned, that only the two-description case is evaluated in this thesis.

Since all MD coding systems appear to introduce very high amounts of redundancy, some way of controlling the correlation, and thereby the redundancy, is also an interesting area for further development. By reducing the amount of redundancy in the descriptions, the problems with low compression rates in the MD systems could be reduced. This is discussed in [12].

7.4.5 Quality Measurement

Since all of the quality measurement techniques in this thesis have some form of weakness, it could be interesting to do a survey where a large amount of human observers rank the different images in a given scale.

7.4.6 Implementation

Some improvements in the developed test application could also be done. Compression ratio could be improved by implementing a more dynamic quantization procedure. This could be achieved by scanning through each block to find the minimum and maximum DCT coefficient, and then scale the values down to a more advantageous range. For example, if you wanted to represent each coefficient with 8 bits; the DCT coefficients are then scaled down to the range [0,255]. This requires that some additional constants are to be communicated between the sender and receiver, namely the offset and the scaling factor. This would also require another type of RLC than that given in the JPEG standard, since only the lowest DCT coefficients, and not necessarily those close to zero, are given a zero value after quantization. Thus, the JPEG Huffman tables also become incompatible. This in turn means that additional overheads, in form of the Huffman table, need to be sent. An additional point to make is that in MDPCT the quantization is done after the splitting, so that the original 8×8 JPEG quantization tables do not apply. This could be solved in a more appropriate way than just taking every other two values from the JPEG quantization tables.

When it comes to entropy coding, some improvements could also be performed. Since the MDSQ systems entropy encodes the indices rather than the quantized DCT coefficients, some new, more suitable Huffman tables could be implemented. The range of the indices is smaller than the range of the DC components; and this implies that Huffman tables with shorter code words could be used. This could improve the compression rates in the MDSQ systems.

A disadvantage with the developed test application is that it is not comparable with real networks. However, the advantage of this application is that the problems regarding real networks are avoided, such as instability, transient channel shutdowns or random number of lost packets. The user can then focus exclusively on an image transmission with a specified amount of packet loss. However, an exciting, but comprehensive, development is therefore to design an independent protocol for MD coded image transmissions, such that the developed systems may be tested in a physical transmission system. The developer has to then allow for real network problems, such as congestion, transient channel shutdown etc. that further will give greater possibilities for additional optimization of the MD coding algorithms.

7.4.7 Future Work Summary

The above-mentioned proposals to further work are only a selection of possible extensions to the work presented in this thesis. MD coding appears to have a wide range of possible applications, especially within the continuously increasing collection of multimedia services.

8 Conclusion

This thesis has evaluated MD coding as a source coding technique for the transmission of images in error-prone networks. The problem with packet loss in data transmission has up to now been solved by retransmission of lost packets, or even by simply ignoring the data loss. When packets are dropped in the network because of congestion, retransmission can lead to an even more congested network, and thus add an unacceptable amount of delay to the transmission. MD coding may remedy this, not by requiring retransmission of lost data, but simply make use of the data successfully transmitted. The MD coding systems evaluated in this thesis splits the image data into two correlated descriptions. If one description is lost, the received description can be used for reconstruction.

This thesis began by setting some realistic targets regarding image quality and file size for the MD coding systems, then all the steps in the image compression were described in detail. A comparison system equal to that of the JPEG standard was implemented for evaluating the effectiveness of the MD coding systems. Three different systems based on [7][3] were implemented and tested using three images of different levels of detail.

The results of the tests show that MD coding can improve the reconstructed quality of a transmitted image significantly, subject to imposed data loss, compared to conventional SD coding. The correlation between the descriptions makes it possible to estimate the lost data to some degree. This correlation introduces a certain amount of redundancy in the descriptions, indeed, in the MD coding systems tested in this thesis, the redundancy tends to be too high compared to the targets set in Section 1.4; the low detailed image yields an especially high amount of redundancy. The target was set to be no more than 30 % redundancy, but as much as about 70 % was discovered in the worst of the cases reported here. The redundancy emphasis lies at about 40-50 % depending on the test image and the compression rate. Thus, some form of redundancy control in the MD coding systems is needed. Another plausible solution is to accept a small reduction in quality compared to the SDTC system when both descriptions are received (equivalent to JPEG), but a quality much better when only one description is received in both systems.

MD coding is until today a relatively unknown and uncommonly used coding technique, but it has, through this thesis project, been shown that it is a promising technique for transferring image data in error-prone networks. With today’s enormous growth within, among other things, mobile multimedia services, there will be a considerable need for effective coding and the successful transmission of information. MD coding may have a great utilitarian value to existing systems as well as to new technology.

Abbreviations

BMP	- Bitmap
BPP	- Bits per pixel
CLI	- Common Language Infrastructure
EOB	- End-of-block
CWT	- Continuous Wavelet Transform
D1	- Description One
D2	- Description Two
DB	- Decibel
DCT	- Discrete Cosine Transform
DWT	- Discrete Wavelet Transform
FFT	- Fast Fourier Transform
GIF	- Graphics Interchange Format
GUI	- Graphical User Interface
HVS	- Human Visual System
IDCT	- Inverse Discrete Cosine Transform
JPEG/JPG	- Joint Photographic Experts Group
MDC	- Multiple Description Coding
MDPCT	- Multiple Description by Pairwise Correlating Transform
MDSQ	- Multiple Description Scalar Quantization
MSE	- Mean Square Error
MSSIM	- Mean Structural Similarity Index
OOP	- Object Oriented Programming
PNG	- Portable Network Graphics
PSNR	- Peak Signal To Noise Ratio
RGB	- Red Green Blue
RLC	- Run Length Coding
SDC	- Single Description Coding
SDDDC	- Single Description Double DC
SDTC	- Single Description Transform Coding
UQI	- Universal Quality Index
UML	- Universal Modelling Language
ZRL	- Zero run-length

Bibliography

[1]
V. K. Goyal
"Multiple Description Coding: Compression Meets the Network"
September 2001
Available: http://lcavwww.epfl.ch/~goyal/Papers/Goyal_2001_MD.pdf

[2]
M. Pereira, M. Antonini and M. Barlaud.
"Multiple description image and video coding for wireless channels"
November 2003
Available: http://www.i3s.unice.fr/~am/Articles/EurasipJASP03/ArticleMDC-EURASIP-JASP.pdf

[3]
V. A. Vaishampayan
"Design of multiple description scalar quantizers"
May 1993

[4]
V. A. Vaishampayan
"Design of entropy-constrained multiple description scalar quantizers"
1994

[5]
V. A. Vaishampayan
"Vector quantizer design for diversity systems"
1991

[6]
V.A. Vaishampayan
"Application of Multiple Description codes to image and video transmission over lossy networks"
March 1996
Available: http://www.research.att.com/~vinay

[7]
Y. Wang, M.T. Orchard, and A.R. Reibman
"Multiple description image coding for noisy channels by pairing transform coefficients"
June 1997
Available: http://vision.poly.edu:8080/~yao/MDC/mdc-MMSP.html

[8]
D. M. Chung, Y. Wang
"Multiple Description Image Coding Using Signal Decomposition And Reconstruction Based on Lapped Orthogonal Transforms"
September 1999
Available: http://vision.poly.edu:8080/paper/CSVT99_LOT.pdf

[9]
V. K. Goyal, J. Kovacevic, R. Arean and M. Vertterli
"Multiple Description Transform Coding of Images"
October 1998
Available: http://www.rle.mit.edu/stir/documents/GoyalKAV_ProcICIP1998.pdf

[10]
A. C. Miguel, A. E. Mohr and E. A. Riskin
"SPIHT for Generalized Multiple Description Coding "
October 1999
Available: http://dcl.ee.washington.edu/papers/mmr-icip-99.ps

[11]
W. Jiang and A. Ortega
"Multiple description coding via polyphase transform and selective quantization"
January 1999
Available: http://citeseer.csail.mit.edu/cache/papers/cs/27633/http:zSzzSzsipi.usc.eduzSz~ortegazSz.zSzPaperszSzvcip99_mdc.pdf/jiang99multiple.pdf

[12]
Y. Wang, M. T. Orchard, V. A. Vaishampayan, A. R. Reibman
"Multiple Description Coding Using Pairwise Correlating Transforms"
January 2001
Available: http://vision.poly.edu:8080/paper/mdc_ip.pdf

[13]
S. D. Servetto, K. Ramchandran, V. A. Vaishampayan, and K. Nahrstedt
"Multiple description wavelet based image coding"
October 1998
Available: http://citeseer.ist.psu.edu/cache/papers/cs/861/ftp:zSzzSzftp.ifp.uiuc.eduzSzpubzSzservettozSzpaperszSzmdImages:tIP.pdf/servetto98multiple.pdf

[14]
R. B. Johnson
"Quantitative, Qualitative, and Mixed Research"
Available: http://www.southalabama.edu/coe/bset/johnson/dr_johnson/lectures/lec2.htm

[15]
Bryman
"Social Research Strategies"
Available: http://www.oup.com/uk/booksites/content/0199264465/bryman_ch01.pdf

[16]
P.E. Pedersen, AUC
"Kvantitative undersøkelsesopplegg"
August 2004

[17]
"The Qualitative Paradigm"
Available: http://www.computing.dcu.ie/~hruskin/RM2.htm
Based on:
John Crosswell
"Research Design: Qualitative and quantitative approaches"

[18]
D. Hill
"Introduction to JPEG"
September 1997
Available: http://www-pg.umds.ac.uk/d.hill/teaching/rsc0/internetIntro/jpeg.html

[19]
DCube Software Technologies
“JPEG Baseline Compression”
June 2002
Available: http://www.funducode.com/freec/Fileformats/format3/format3c.htm

[20]
B. Brower
”Image Compression Basics”
March 1999
Available: http://164.214.2.51/ntb/2002SICS/Comp_Tutorial_Apr02_pt2.PDF

[21]
Wikipedia
“Chroma subsampling”
February 2005
Available: http://en.wikipedia.org/wiki/4:2:0

[22]
D. Munsil and S. Spears
“The Chroma Upsampling Error and The 4:2:0 Interlaced Chroma Problem”
January 2003
Available: http://www.hometheaterhifi.com/volume_8_2/dvd-benchmark-special-report-chroma-bug-4-2001.html

[23]
J. Burg and Y. L. Wong
“JPEG Compression”
Spring 2003
Available: http://www.cs.wfu.edu/%7Eburg/Courses/Spring04/CSC361-661/course-materials/JPEGCompression.ppt

[24]
G. K. Wallace
“The JPEG still picture compression standard”
December 1991
Available: http://white.stanford.edu/~brian/psy221/reader/Wallace.JPEG.pdf

[25]
International Telecommunication Union (ITU)
“Terminal Equipment and Protocols for Telematic Services”
September 1992
Available: http://www.w3.org/Graphics/JPEG/itu-t81.pdf

[26]
M. Donovan
“MPEGBuilder1SourceCode – C# Project”
January 2004
Available: http://www.codeproject.com/cs/media/MPEGFunction.asp

[27]
N. Kingsbury
”JPEG Entropy Coding”
May 2003
Available: http://cnx.rice.edu/content/m11096/latest/

[28]
Wikipedia
”JPEG Encoding”
February 2005
Available: http://en.wikipedia.org/wiki/JPEG#Encoding

[29]
Wikipedia
”Huffman coding”
February 2005
Available: http://en.wikipedia.org/wiki/Huffman_coding

[30]
Y.T. Chang
“Wavelet basics”
1995

[31]
S. Shen
”Discrete Wavelet Transform (DWT)”
May 2004
Available: http://www.csee.umbc.edu/~pmundur/courses/CMSC691M-04/sharon-DWT.ppt

[32]
Wikipedia
“JPEG 2000”
February 2005
Available: http://en.wikipedia.org/wiki/JPEG_2000

[33]
K. R. Persons, W. J. Charboneau, E.M. James et al.
“Ultrasound grayscale image compression with JPEG and wavelet techniques”
February 2000

[34]
Graphics Software
“JPEG Format - About JPEG and JPEG 2000 File Formats”
May 2005
Available: http://graphicssoft.about.com/od/formatsjpeg

[35]
P. L. Dragotti
“Wavelet Footprints and Frames for Signal Processing and Communication”
April 2002

[36]
Matthias Pätzold, AUC
”Antenna Diversity”
2004

[37]
C. L. Heng
“Image Compression: JPEG”
May 1997
Available: http://pascalzone.amirmelamed.co.il/Graphics/JPEG/JPEG.htm

[38]
Z. Wang and A. C. Bovik
"A Universal Image Quality Index"
March 2002
Available: http://www.cns.nyu.edu/~zwang/files/papers/quality_2c.pdf

[39]
Z. Wang and A. C. Bovik
"A Universal Image Quality Index"
April 2004
Available: http://filter.cs.ccu.edu.tw/courses/image_processing_pu(2004)/slides/data/2004-4-21/2_A%20Universal%20Image%20Quality%20Index.ppt

[40]
Wikipedia
"Peak signal to noise ratio"
February 2005
Available: http://en.wikipedia.org/wiki/PSNR

[41]
Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli
"Image Quality Assessment: From Error Visibility to Structural Similarity"
April 2004
Available: http://www.cns.nyu.edu/pub/eero/wang03-reprint.pdf

[42]
"The Visual Studio .NET Framework"
Available: http://www.wilsonmar.com/img/fig/dotnetfw.gif

[43]
Z. Xiong, K. Ramchandran, M. T. Orchard and Y. Q. Zhang
"A Comparative Study of DCT- and Wavelet-Based Image Coding"
August 1999
Available: http://research.microsoft.com/china/papers/Comparative_Study_DCT_WaveletBased_Image_Coding.pdf

[44]
Scott Clark
"Compression - Optimizing Web Graphics"
February 2000
Available: http://www.webreference.com/dev/graphics/compress.html

Appendix A: Huffman Tables

Table A-1. Cat - Range

Cat	Range
0	0
1	–1,1
2	–3,–2,2,3
3	–7..–4,4..7
4	–15..–8,8..15
5	–31..–16,16..31
6	–63..–32,32..63
7	–127..–64,64..127
8	–255..–128,128..255
9	–511..–256,256..511
10	–1 023..–512,512..1 023
11	–2 047..–1 024,1 024..2 047

Table A-2. DC Luminance Codes

Category	Code length	Code word
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

Table A-3. DC Chrominance Codes

Category	Code length	Code word
0	2	00
1	2	01
2	2	10
3	3	110
4	4	1110
5	5	11110
6	6	111110
7	7	1111110
8	8	11111110
9	9	111111110
10	10	1111111110
11	11	11111111110

Table A-4. AC Luminance Codes

Run/Size	Code length	Code word
0/0 (EOB)	4	1010
0/1	2	00
0/2	2	01
0/3	3	100
0/4	4	1011
0/5	5	11010
0/6	7	1111000
0/7	8	11111000
0/8	10	1111110110
0/9	16	1111111110000010
0/A	16	1111111110000011
1/1	4	1100
1/2	5	11011
1/3	7	1111001
1/4	9	111110110
1/5	11	11111110110
1/6	16	1111111110000100
1/7	16	1111111110000101
1/8	16	1111111110000110
1/9	16	1111111110000111
1/A	16	1111111110001000
2/1	5	11100
2/2	8	11111001
2/3	10	1111110111
2/4	12	111111110100
2/5	16	1111111110001001
2/6	16	1111111110001010
2/7	16	1111111110001011
2/8	16	1111111110001100
2/9	16	1111111110001101
2/A	16	1111111110001110
3/1	6	111010
3/2	9	111110111
3/3	12	111111110101
3/4	16	1111111110001111
3/5	16	1111111110010000
3/6	16	1111111110010001
3/7	16	1111111110010010
3/8	16	1111111110010011
3/9	16	1111111110010100
3/A	16	1111111110010101

Run/Size	Code length	Code word
4/1	6	111011
4/2	10	1111111000
4/3	16	1111111110010110
4/4	16	1111111110010111
4/5	16	1111111110011000
4/6	16	1111111110011001
4/7	16	1111111110011010
4/8	16	1111111110011011
4/9	16	1111111110011100
4/A	16	1111111110011101
5/1	7	1111010
5/2	11	11111110111
5/3	16	1111111110011110
5/4	16	1111111110011111
5/5	16	1111111110100000
5/6	16	1111111110100001
5/7	16	1111111110100010
5/8	16	1111111110100011
5/9	16	1111111110100100
5/A	16	1111111110100101
6/1	7	1111011
6/2	12	111111110110
6/3	16	1111111110100110
6/4	16	1111111110100111
6/5	16	1111111110101000
6/6	16	1111111110101001
6/7	16	1111111110101010
6/8	16	1111111110101011
6/9	16	1111111110101100
6/A	16	1111111110101101
7/1	8	11111010
7/2	12	111111110111
7/3	16	1111111110101110
7/4	16	1111111110101111
7/5	16	1111111110110000
7/6	16	1111111110110001
7/7	16	1111111110110010
7/8	16	1111111110110011
7/9	16	1111111110110100
7/A	16	1111111110110101
8/1	9	111111000
8/2	15	111111111000000

Run/Size	Code length	Code word
8/3	16	1111111110110110
8/4	16	1111111110110111
8/5	16	1111111110111000
8/6	16	1111111110111001
8/7	16	1111111110111010
8/8	16	1111111110111011
8/9	16	1111111110111100
8/A	16	1111111110111101
9/1	9	111111001
9/2	16	1111111110111110
9/3	16	1111111110111111
9/4	16	1111111111000000
9/5	16	1111111111000001
9/6	16	1111111111000010
9/7	16	1111111111000011
9/8	16	1111111111000100
9/9	16	1111111111000101
9/A	16	1111111111000110
A/1	9	111111010
A/2	16	1111111111000111
A/3	16	1111111111001000
A/4	16	1111111111001001
A/5	16	1111111111001010
A/6	16	1111111111001011
A/7	16	1111111111001100
A/8	16	1111111111001101
A/9	16	1111111111001110
A/A	16	1111111111001111
B/1	10	1111111001
B/2	16	1111111111010000
B/3	16	1111111111010001
B/4	16	1111111111010010
B/5	16	1111111111010011
B/6	16	1111111111010100
B/7	16	1111111111010101
B/8	16	1111111111010110
B/9	16	1111111111010111
B/A	16	1111111111011000
C/1	10	1111111010
C/2	16	1111111111011001
C/3	16	1111111111011010
C/4	16	1111111111011011

Run/Size	Code length	Code word
C/5	16	1111111111011100
C/6	16	1111111111011101
C/7	16	1111111111011110
C/8	16	1111111111011111
C/9	16	1111111111100000
C/A	16	1111111111100001
D/1	11	11111111000
D/2	16	1111111111100010
D/3	16	1111111111100011
D/4	16	1111111111100100
D/5	16	1111111111100101
D/6	16	1111111111100110
D/7	16	1111111111100111
D/8	16	1111111111101000
D/9	16	1111111111101001
D/A	16	1111111111101010
E/1	16	1111111111101011
E/2	16	1111111111101100
E/3	16	1111111111101101
E/4	16	1111111111101110
E/5	16	1111111111101111
E/6	16	1111111111110000
E/7	16	1111111111110001
E/8	16	1111111111110010
E/9	16	1111111111110011
E/A	16	1111111111110100
F/0 (ZRL)	11	11111111001
F/1	16	1111111111110101
F/2	16	1111111111110110
F/3	16	1111111111110111
F/4	16	1111111111111000
F/5	16	1111111111111001
F/6	16	1111111111111010
F/7	16	1111111111111011
F/8	16	1111111111111100
F/9	16	1111111111111101
F/A	16	1111111111111110

Table A-5. AC Chrominance Codes

Run/Size	Code length	Code word
0/0 (EOB)	2	00
0/1	2	01
0/2	3	100
0/3	4	1010
0/4	5	11000
0/5	5	11001
0/6	6	111000
0/7	7	1111000
0/8	9	111110100
0/9	10	1111110110
0/A	12	111111110100
1/1	4	1011
1/2	6	111001
1/3	8	11110110
1/4	9	111110101
1/5	11	11111110110
1/6	12	111111110101
1/7	16	1111111110001000
1/8	16	1111111110001001
1/9	16	1111111110001010
1/A	16	1111111110001011
2/1	5	11010
2/2	8	11110111
2/3	10	1111110111
2/4	12	111111110110
2/5	15	111111111000010
2/6	16	1111111110001100
2/7	16	1111111110001101
2/8	16	1111111110001110
2/9	16	1111111110001111
2/A	16	1111111110010000
3/1	5	11011
3/2	8	11111000
3/3	10	1111111000
3/4	12	111111110111
3/5	16	1111111110010001
3/6	16	1111111110010010
3/7	16	1111111110010011
3/8	16	1111111110010100
3/9	16	1111111110010101
3/A	16	1111111110010110
4/1	6	111010

Run/Size	Code length	Code word
4/2	9	111110110
4/3	16	1111111110010111
4/4	16	1111111110011000
4/5	16	1111111110011001
4/6	16	1111111110011010
4/7	16	1111111110011011
4/8	16	1111111110011100
4/9	16	1111111110011101
4/A	16	1111111110011110
5/1	6	111011
5/2	10	1111111001
5/3	16	1111111110011111
5/4	16	1111111110100000
5/5	16	1111111110100001
5/6	16	1111111110100010
5/7	16	1111111110100011
5/8	16	1111111110100100
5/9	16	1111111110100101
5/A	16	1111111110100110
6/1	7	1111001
6/2	11	11111110111
6/3	16	1111111110100111
6/4	16	1111111110101000
6/5	16	1111111110101001
6/6	16	1111111110101010
6/7	16	1111111110101011
6/8	16	1111111110101100
6/9	16	1111111110101101
6/A	16	1111111110101110
7/1	7	1111010
7/2	11	11111111000
7/3	16	1111111110101111
7/4	16	1111111110110000
7/5	16	1111111110110001
7/6	16	1111111110110010
7/7	16	1111111110110011
7/8	16	1111111110110100
7/9	16	1111111110110101
7/A	16	1111111110110110
8/1	8	11111001
8/2	16	1111111110110111
8/3	16	1111111110111000

Run/Size	Code length	Code word
8/4	16	1111111110111001
8/5	16	1111111110111010
8/6	16	1111111110111011
8/7	16	1111111110111100
8/8	16	1111111110111101
8/9	16	1111111110111110
8/A	16	1111111110111111
9/1	9	111110111
9/2	16	1111111111000000
9/3	16	1111111111000001
9/4	16	1111111111000010
9/5	16	1111111111000011
9/6	16	1111111111000100
9/7	16	1111111111000101
9/8	16	1111111111000110
9/9	16	1111111111000111
9/A	16	1111111111001000
A/1	9	111111000
A/2	16	1111111111001001
A/3	16	1111111111001010
A/4	16	1111111111001011
A/5	16	1111111111001100
A/6	16	1111111111001101
A/7	16	1111111111001110
A/8	16	1111111111001111
A/9	16	1111111111010000
A/A	16	1111111111010001
B/1	9	111111001
B/2	16	1111111111010010
B/3	16	1111111111010011
B/4	16	1111111111010100
B/5	16	1111111111010101
B/6	16	1111111111010110
B/7	16	1111111111010111
B/8	16	1111111111011000
B/9	16	1111111111011001
B/A	16	1111111111011010
C/1	9	111111010
C/2	16	1111111111011011
C/3	16	1111111111011100
C/4	16	1111111111011101
C/5	16	1111111111011110

Run/Size	Code length	Code word
C/6	16	1111111111011111
C/7	16	1111111111100000
C/8	16	1111111111100001
C/9	16	1111111111100010
C/A	16	1111111111100011
D/1	11	11111111001
D/2	16	1111111111100100
D/3	16	1111111111100101
D/4	16	1111111111100110
D/5	16	1111111111100111
D/6	16	1111111111101000
D/7	16	1111111111101001
D/8	16	1111111111101010
D/9	16	1111111111101011
D/A	16	1111111111101100
E/1	14	11111111100000
E/2	16	1111111111101101
E/3	16	1111111111101110
E/4	16	1111111111101111
E/5	16	1111111111110000
E/6	16	1111111111110001
E/7	16	1111111111110010
E/8	16	1111111111110011
E/9	16	1111111111110100
E/A	16	1111111111110101
F/0 (ZRL)	10	1111111010
F/1	15	111111111000011
F/2	16	1111111111110110
F/3	16	1111111111110111
F/4	16	1111111111111000
F/5	16	1111111111111001
F/6	16	1111111111111010
F/7	16	1111111111111011
F/8	16	1111111111111100
F/9	16	1111111111111101
F/A	16	1111111111111110

Appendix B: Coding and Decoding Sequence

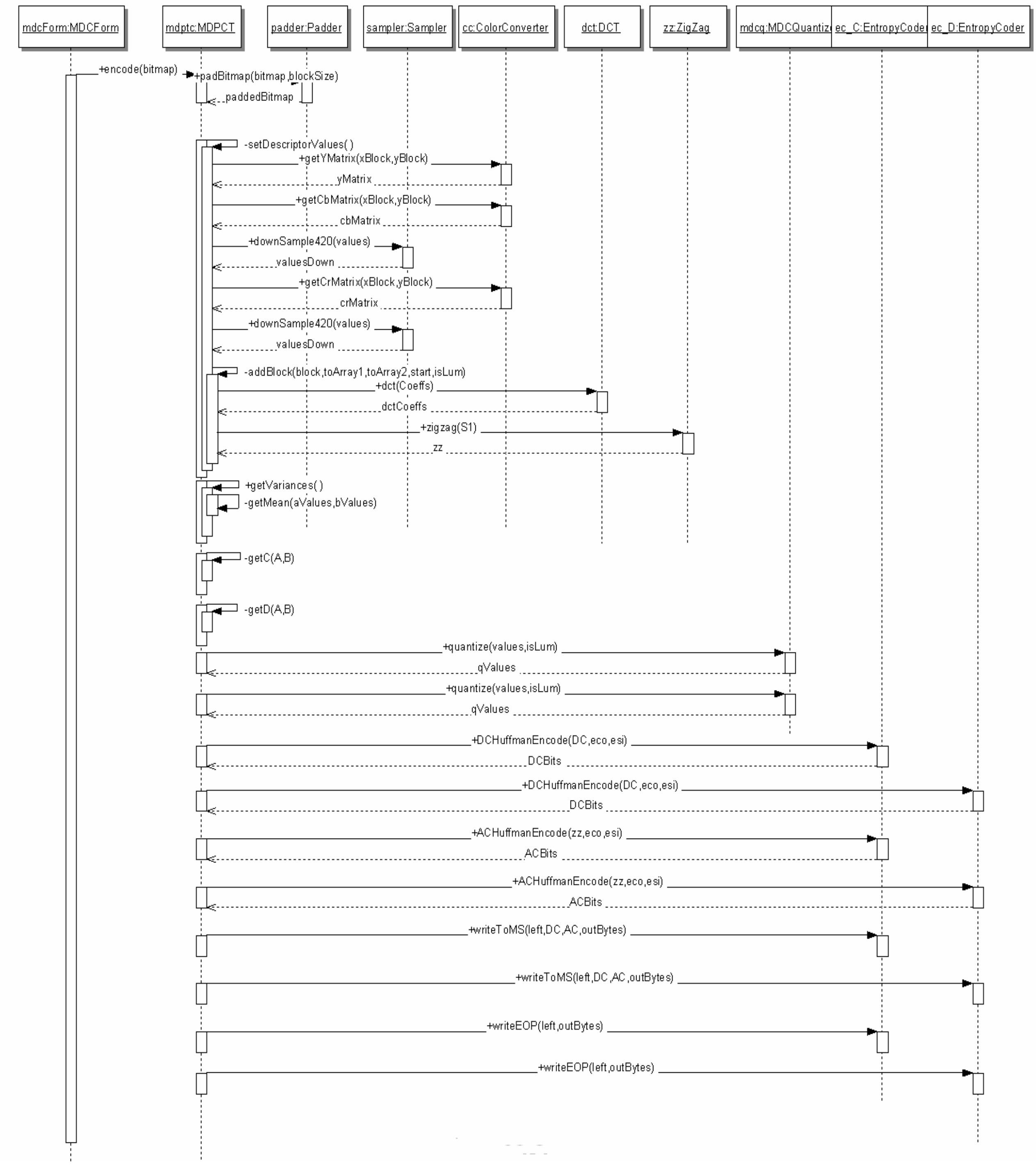


Figure B-1: The MDPCT encoding sequence

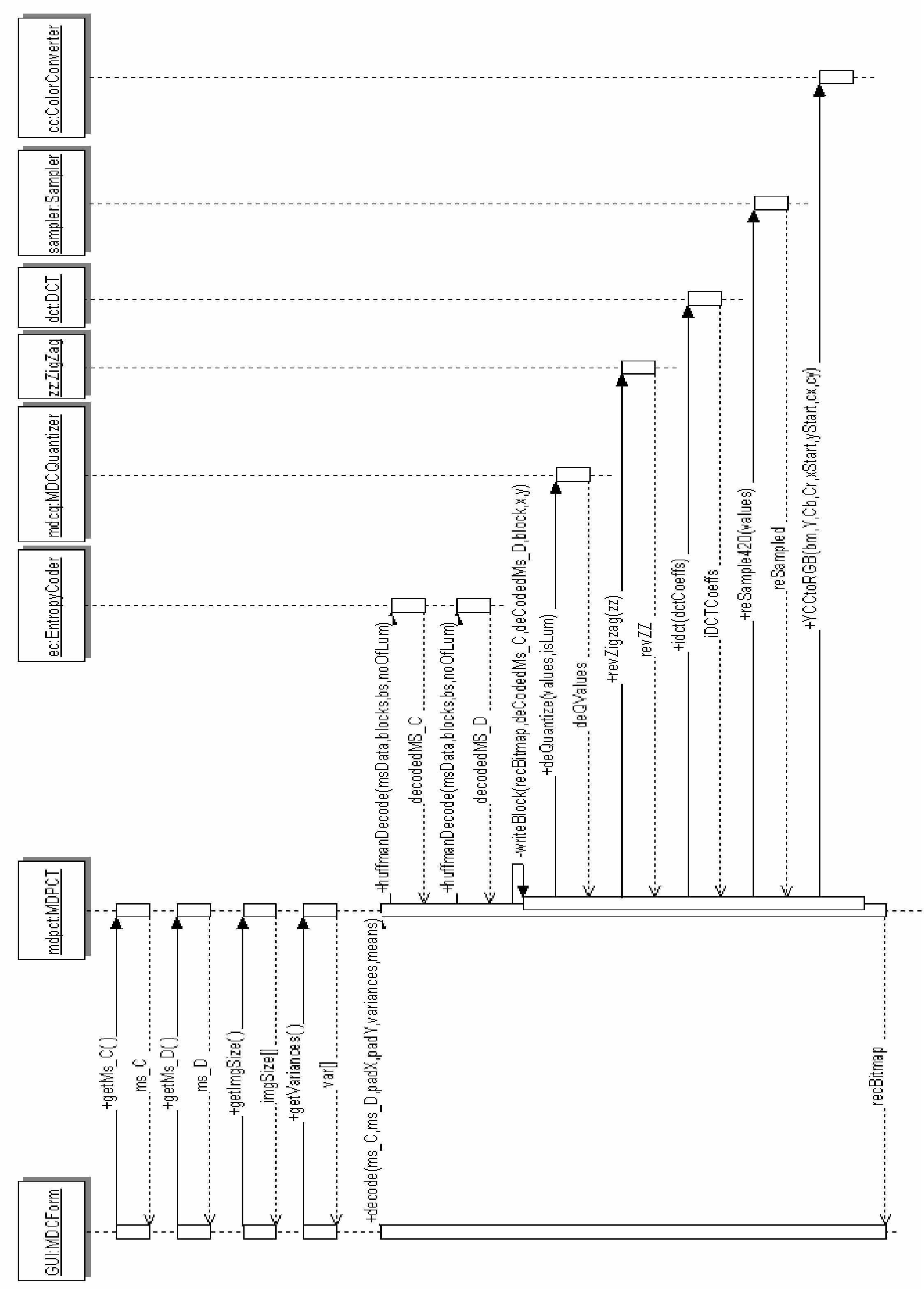


Figure B-2: The MDPCT decoding sequence

Appendix C: Summary of System Performances

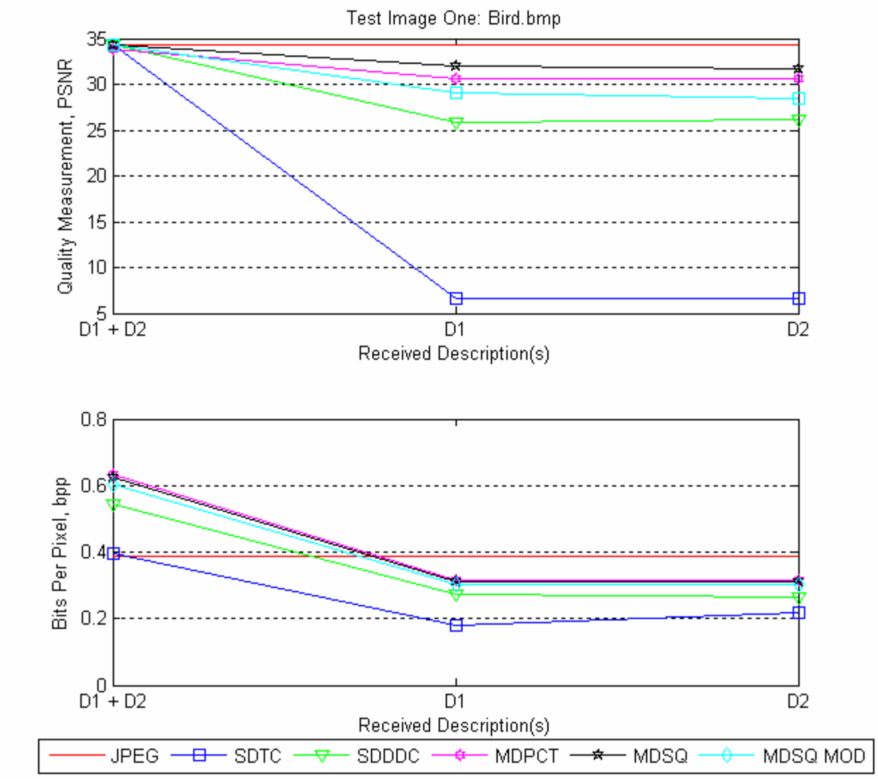


Figure C-1: Summary - Bird.bmp

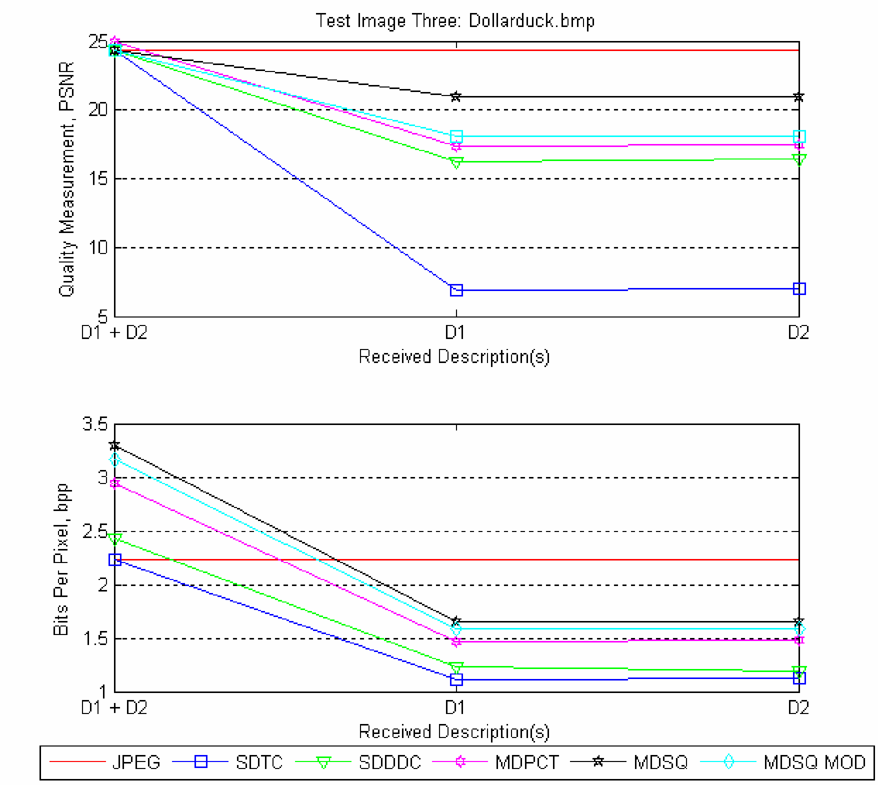


Figure C-2: Summary - Dollarduck.bmp

Appendix D: One versus Both Descriptions Comparison

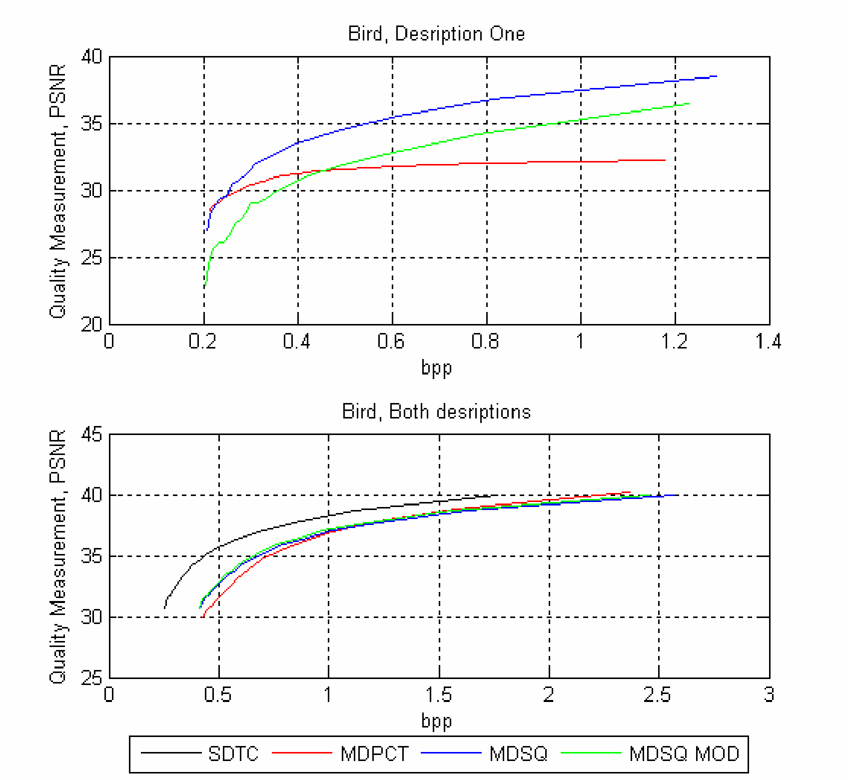


Figure D-1: Bird - One versus both descriptions

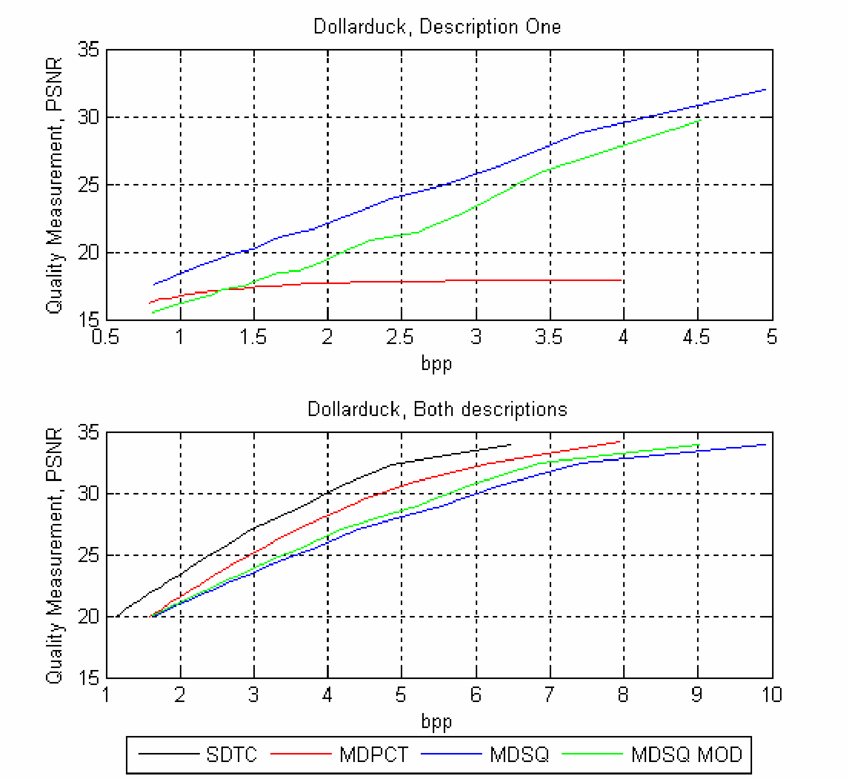


Figure D-2: Dollarduck - One versus both descriptions

Appendix E: C# source code and test images

The C# source code and test images can be found on the CD-ROM attached to this report.