IMAGE/VIDEO COMPRESSION AND QUALITY ASSESSMENT BASED ON WAVELET TRANSFORM

DISSERTATION

Presented in Partial Fulfillment of the Requirements for

the Degree Doctor of Philosophy in the

Graduate School of The Ohio State University

By

Zhigang Gao, M.S.

* * * * *

The Ohio State University

2007

Dissertation Committee:

Prof. Yuan F. Zheng, Adviser

Prof. Ashok Krishnamurthy

Prof. Phil Schniter

Approved by

Adviser

Graduate Program in
Electrical and Computer
Engineering

ABSTRACT

Because of the contradiction of the vast data size of raw digital images and videos and the limited transmission bandwidth and storage space, it is essential to develop compression methodologies with high compression ratio and good reconstructed quality. It is also important to develop quality metrics which are consistent with human vision and easy to calculate. The spatial-frequency localization and multi-resolution capabilities of the wavelet transform make it a natural means of signal representation. This work investigates the advantages of the wavelet transform and focuses on the following research topics:

1. An image quality metric that assesses the quality of an image in the wavelet domain.

2. A quality constrained compression algorithm that compresses an image to a desired visual quality.

3. An innovative DWT-based temporal filtering scheme that achieves high compression ratio and reduces the ghost effect without motion estimation.

4. A virtual sub-object video coding scheme that is suitable for applications with static background.

This is dedicated to ... my parents and wife ...

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor Prof. Yuan F. Zheng for welcoming me into his Wavelets group, which is a truly unique research environment. I appreciate his guidance and the numerous discussions that we had. His diligence and passion on scientific research really impressed me. I learned a lot from him.

I am grateful to Prof. Ashok Krishnamurthy and Prof. Phil Schniter for accepting to be part of the committee, and for their feedbacks regarding my dissertation.

I enjoyed working with my colleagues in the Wavelets group. It was a real blessing to be able to work with these intelligent and hard working people. I worked together closely with Dr. Jianyu Dong, Dr. Yi Liu, Dr. Eric Balster who helped me shape and realize many ideas. Furthermore, I thank all the friends who participated in my subjective experiments for their time and patience.

VITA

Febuary 14, 1974	Born in Taiyuan, P. R. China
June 1997	B.S. in Electronics and Information Systems, Beijing University
March 2002	M.S. in Electrical and Computer Engineering, The Ohio State University
January 2002 - December 2005	Graduate Research Associate, The Ohio State University.
April 2002 - present	Ph.D. candidate in Electrical and Computer Engineering, The Ohio State University.
June 2006 - present	Hardware Engineer, Cisco Systems.

PUBLICATIONS

Research Publications

Zhigang Gao and Yuan F. Zheng, "An innovative temporal wavelet filtering scheme for video coding", *ICIP'2005*, vol. 3, pp. 205-208, September 2005.

Zhigang Gao and Yuan F. Zheng, "Motion optimized spatial-temporal video coding based on wavelet transform", *ACSSC'2004*, vol. 2, pp. 1718-1722, November 2004.

Zhigang Gao and Yuan F. Zheng, "Variable quantization in subbands for optimal compression using wavelet transform", *SCI'2003*, vol. 2, pp. 321-326, July 2003.

Chao He, Jianyu Dong, Yuan F. Zheng and Zhigang Gao, "Optimal 3-D coefficient tree structure for 3-D wavelet video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 961-972, October 2003.

Zhigang Gao and Yuan F. Zheng, "Quality constrained compression using dwt based image quality metric," submitted to *IEEE Transactions on Circuits and Systems for Video Technology*, April 2007.

FIELDS OF STUDY

Major Field: Electrical and Computer Engineering

Studies in Computer Engineering: Prof. Yuan F. Zheng

TABLE OF CONTENTS

Page

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Motivations

Humans are highly visual creatures. We are well adapted to the visual world around us and rely heavily on visual information for our daily activities. As our world is becoming more digitalized every day, it is not surprising that digitalized images and videos are becoming more and more common and wide-spread. In light of this, optimizing the performance of digital systems that capture, display, store or transmit images or videos becomes one of the most important challenges.

With the vast amount of digital images and videos, which is still increasing rapidly, the transmission bandwidth and storage spaces become the bottleneck. Compression is the natural solution for this problem. To achieve a satisfactory compression ratio, the lossy compression tools have to be used, which will cause degradation of visual quality of the compressed images or videos. So there is contradiction between compression efficiency and visual quality. People working on image and video compression are always looking for solutions that can achieve relatively higher compression ratio than existing methods while maintaining a comparable visual quality. Since humans are the end users of these visual data, the visual quality should be judged from the

human point of view. A reliable quality metric that is consistent with human vision will not only have its own applications, but also help the development of compression technologies.

1.1.1 Image Quality Metric

Quality assessment metrics should evaluate the visual quality of images and videos with subject to the subjective ranking (human vision). There is no way to evaluate the performance of a compression tool without reliable quality measurement. Having the images or videos viewed by human observers is one way to obtain reliable ratings of the quality of them. While these experiments are the closest we can reach to the truth about perceived quality, they are too complex, time-consuming and consequently expensive. Hence, they are often only used as bench marks for research and development purpose.

While looking for faster alternatives, the researchers in the field of image quality assessment have turned to simple pixel error measures such as mean squared error (MSE) or peak signal-to-noise ratio (PSNR). However, these simple measures operate solely on a pixel-by-pixel basis in the spatial domain and neglect the functioning mechanism of HVS. Therefore, their predictions of the image quality often do not agree well with the perceived quality by humans.

These problems stimulated the study of human vision and visual quality metrics in recent years. The human vision system (HVS) is a very complex system that is still not quite understood by researchers. But some low level characteristics of it have been studied for a long time and proven to be reliable in directing the development of visual technologies. A great deal of effort has been made to develop image quality

metrics that are based on the human visual system. While some metrics yield decent results, most of them are not always consistent with human perceived quality and are sometimes limited to very specific applications. Furthermore, these metrics tend to be very complicated for implementation. Besides, they are all measured in the spatial domain while the compression is performed in the frequency domain, which makes it very difficult to adjust the quality during the procedure of compressing.

With this in mind, we thought it was necessary to develop an image quality metric based on the concepts of the human visual system in the wavelet domain, which is easy to implement, consistent with HVS ranking, and able to measure the post-compression quality without reconstructing the compressed image.

1.1.2 Compression

Effective image and video compression techniques have been two very active research areas in the past two decades. It is essential to develop compression methodologies which can both produce high compression ratios and preserve good reconstructed quality.

Although Discrete Cosine Transform (DCT) based compression techniques are suitable for moderate data compression of image and video signals, the quality of reconstructed signals becomes poor when compression ratios are high. The multiresolution capabilities of the Discrete Wavelet Transform (DWT) in both the spatial and frequency domains make it a more natural means of signal representation. The DWT based compression techniques have shown to outperform DCT-based solutions, especially at high compression ratios. JPEG released the JPEG2000 standard, which is a wavelet-based image compression technology that is slowly but sturdily replacing

the original DCT-based JPEG standard. Also, as extensions of the wavelet based image compression techniques, 3-D wavelet compression techniques have shown great potential in the video compression field in the recent years.

In light of this, we chose DWT based image and video compression as one of our research areas. We noticed that image compression is usually treated as a bit-rate constrained problem, i.e., compression ratio is on the top of consideration while quality is secondary. Since the features of images may vary significantly, image qualities can be very different for the same bit-rate. There are applications that want the visual qualities of the compressed images be constrained in an acceptable range, where bit-rate constant compression is not desired. So we decided to develop a compression method which prioritizes the quality (*quality constrained compression*), which compresses images to a desired visual quality.

In the field of video compression, the motion estimation/compensation scheme has been playing a key role for a long time. The block-based motion estimation is the dominant approach, but the object-based approach is gaining its share due to the need of object-based/content-based video applications. We noticed that the motion vectors and shape maps undermine the compression efficiency. So we conducted researches on video compression in two new directions to overcome this problem: one is a temporal filtering algorithm that does not use motion vectors and the other is an object-based coding method that does not need to code shape information.

1.2 Approaches

This work covers a few areas in image and video technology using the wavelet transform, including image quality assessment, quality constrained image compression and 3-D wavelet video compression.

The most important tool this work based on is the DWT. This work searches for a better way to represent the visual signals and describe the HVS concepts in the domain of wavelet transform. This is because not only that the spatial-temporal multi-resolution property of wavelet is perfect to describe the visual signals, but also that DWT has achieved great success in image and video technology in the recent years. This dissertation further investigated of the application of DWT in image and video technology and took the following research approaches.

1. HVS based image quality metrics tend to be complex and hard to implement, this work studied the concepts of HVS and intended to find an accurate and simple HVS based solution. This solution should be able to cover the most important features of HVS and wavelet friendly, i.e., suitable to be implemented in the wavelet domain.

2. One important feature of quality constrained compression is real-time ready, i.e., the quality of the compressed image should be able to be controlled or adjusted during the compression without reconstructing the image. With the help of the wavelet domain image quality metric, this becomes possible since the lossy coding that determines the image quality is in the wavelet domain.

3. Video compression is more complex than image compression because it has to deal with the temporal redundancy in addition. How to remove the temporal

redundancy is the most important issue in video compression. This work was trying to explore this problem in new directions. The first one is the sub-grouping transform algorithm that breaks down the uniform transformation of video frames and provides the flexible pixel-based temporal transform. The second one is a hybrid of block-based and object-based motion estimation, and combines shape adaptive and regular transforms. Both of them avoided the computation intensive motion estimation and compensation approach. But they are able to be used with an additional sophisticated motion estimation and compensation algorithm.

1.3 Contributions

The major contributions of this dissertation can be summarized as follows:

1. A new quality metric in the wavelet domain called WNMSE is proposed. WNMSE is consistent with the human judgment of visual quality, simple to implement, and able to estimate the quality of an image during the compression.

2. A real-time quality constrained compression algorithm called QCSQ is proposed. QCSQ is based on the relationship among the statistic features, quantization step-sizes, and WNMSE value of a compressed image. It can determine the quantization step-sizes for all the wavelet subbands of a DWT decomposed image and compress this image to a desired visual quality accurately.

3. An innovative temporal wavelet filtering scheme that is not dependent on motion-estimation and motion vectors is proposed. Compared with other video coding

algorithms with motion-compensation, the so-called Sub-Grouping Transformation (SGT) algorithm has nearly no overhead bits.

4. A 3-D virtual object coding scheme that exploits the strengths of both block-based and object-based motion estimation, and combines the regular and shape adaptive transforms. By defining the base frame, spatially resizing VOPs, and dividing VOPs into sub-objects, this method can achieve both motion estimation accuracy and compression efficiency with only one motion trajectory for the video object without the shape coding overhead.

1.4 Outline

This dissertation is organized as follows: the human vision system, as the main foundation of the next chapter, is introduced in Chapter 2. The proposed image quality metric and the quality constrained compressed based on it are in Chapter 3. Chapter 4 presents the pixel-based sub-grouping transform algorithm for video coding. The virtual sub-object based video coding is in Chapter 5. Finally, Chapter 6 concludes the dissertation. The detailed algorithms of QCSQ are listed in the Appendices.

CHAPTER 2

HUMAN VISION SYSTEM

2.1 Introduction

Vision is the most essential one of our senses. As a matter of fact, 80-90 percent of all neurons in the human brain are estimated to be devoted to vision signal processing [1]. This indicates the extreme complexity of the human visual system, which can be divided into two major components: the eyes that capture light and convert it into signals that can be understood by the nervous system, and the nervous pathways in the brain, along which these signals are transmitted and processed.

For people who are working on image and video technologies, it is very important to study the characteristics of the human visual system and apply it into research. Although the current knowledge about the human visual system is still limited, there are aspects of the human visual system that are relevant to image and video processing and can be very helpful to research on image and video techniques.

This chapter will discuss the structure and functions of the human visual system as well as a number of properties of it that are of particular interest of our research.

2.1.1 Optics: Orientation and Color Dependent

From an optical point of view, the eye is the equivalent of a camera, which comprises a system of lenses and a variable aperture to focus images on the light-sensitive retina. The optical system of the human eye is composed of the cornea, the aqueous humor, the lens, and the vitreous humor. Because the cornea is not perfectly symmetric, the optical properties of the eye are orientation dependent. Therefore it is impossible to perfectly focus stimuli of all orientations simultaneously.

The properties of the eye's optics, most importantly the refractive indexes of the optical elements, vary with wavelength. This means that it is impossible to focus all wavelengths simultaneously. It is evident that the retinal image contains only poor spatial details for wavelengths far from away the center wavelength (or color). This tendency towards monochromacy (total color blindness) becomes even more pronounced with increasing luminance.

2.1.2 Spatial Resolution: Limited and Luminance Dependent

The optics of the eye project images of the outside world on the retina, the neural tissue at the back of the eye, where the layer of light sensitive photoreceptors is located. The photoreceptors are specialized neurons that convert the light energy into signals that can be interpreted by the brain. The size and spacing of the photoreceptors determine the maximum spatial resolution of the human visual system. There are two different types of photoreceptors called rods and cones. Rods are responsible for low light level vision, while cones for high light level vision.

Rods are very sensitive light detectors with poor visual resolution. This is due to the fact that signals from many rods converge onto a single neuron, which improves sensitivity but reduces resolution. The opposite is true for the cones that several neurons encode the signal from each cone.

2.1.3 Sensitivity: Selective to Frequency, Color, Velocity, Orientation, and Phase

The visual cortex is responsible for all higher-level aspects of vision. There is an enormous variety of cells in the visual cortex. A particular cell may respond strongly to patterns of a certain orientation or to motion in a certain direction. Similarly, there are cells tuned to particular frequencies, colors, velocities, etc. This neuronal selectivity is thought to be at the heart of the multi-channel organization of human vision.

2.2 Contrast Sensitivity

The human visual system is capable of adapting to an enormous range of light intensities. The response of the human visual system depends much less on the absolute luminance than on the relation of its local variations to the surrounding luminance. This property is known as Weber-Fechner law. Contrast is a measure of this relative variation of luminance. Mathematically, Weber contrast can be expressed as

$$C^W = \frac{L - L_b}{L_b}, \tag{2.1}$$

where L is the luminance of the pixel of interest and L_b is the background luminance. The threshold contrast, i.e. the minimum contrast necessary for an observer to detect a change in intensity, is a function of background luminance. It remains nearly

constant from faint lighting to daylight, which is indeed the luminance range typically encountered in most image processing applications. Evidently, the Weber-Fechner law is only an approximation of the actual contrast.

The threshold contrast depends to a great extent on the stimulus characteristics, most importantly its color as well as its spatial and temporal frequency. Contrast sensitivity is defined as the inverse of the contrast threshold. Contrast sensitivity functions (CSF) are generally used to quantify these dependencies. In these CSF measurements, the contrast of periodic stimuli with varying frequencies is defined as the Michelson contrast [2]:

$$C^M = \frac{L_{max} - L_{min}}{L_{max} + L_{min}}, \tag{2.2}$$

where L_{min} and L_{max} are the luminance extremes of the pattern.

While the above two definitions are good predictors of perceived contrast for simple stimuli, they fail when stimuli become more complex and cover a wider frequency range. It is also evident that none of these simple global definitions is appropriate for measuring contrast in natural images. This is because a few very bright or very dark points would determine the contrast of the whole image, whereas actual human contrast perception varies with the local average luminance. In order to address these issues, Peli proposed a local band-limited contrast in [3]:

$$C^P{}_j(x, y) = \frac{\psi_j * L(x, y)}{\phi_j * L(x, y)}, \tag{2.3}$$

where ψ_j is a band-pass filter at level j of a filter bank, ϕ_j is the corresponding low-pass filter, and $L(x, y)$ is the luminance at (x, y).

In [4], Lubin modified Peli's contrast definition in an image quality metric based on a multi-channel model of the human visual system:

$$C^L{}_j(x,y) = \frac{(\phi_j - \phi_{j+1}) * L(x,y)}{\phi_{j+2} * L(x,y)}. \tag{2.4}$$

The differences between C^P and C^L are most pronounced for higher-frequency bands. The lower the frequency, the more spatially uniform the low-pass band in the denominator will become in both, finally approaching the overall luminance mean of the image.

Local contrast as defined above measures contrast only with respect to the local background. This is analogous to the symmetric (in-phase) responses of vision mechanisms. However, a complete description of contrast for complex stimuli has to include the anti-symmetric (quadrature) responses as well [5, 6]. Analytic filters represent an elegant way to achieve this: The magnitude of the analytic filter response, which is the sum of the energy responses of in-phase and quadrature components, exhibits the desired behavior in that it gives a constant response to sinusoidal gratings.

Oriented measures of contrast can still be computed, because the Hilbert transform is well-defined for filters whose angular support is smaller than π. Such contrast measures are useful for many image processing tasks. They can implement a multi-channel representation of low-level vision in accordance with the orientation selectivity of the human visual system and facilitate modeling aspects such as contrast sensitivity and pattern masking. Contrast pyramids have also been found to reduce the dynamic range in the transform domain, which may find interesting applications in image compression [7].

For example, in [4], Lubin applies oriented filtering to $C^L{}_j$ and sums the squares of the in-phase and quadrature responses for each channel to obtain a phase-independent

oriented measure of contrast energy. Using analytic orientation-selective filters $\eta_k(x, y)$, this oriented contrast can be expressed as

$$C^L{}_{jk}(x, y) = |\eta_k * C^L{}_{jk}(x, y)|. \tag{2.5}$$

Alternatively, an oriented pyramid decomposition can be computed first, and contrast can be defined by normalizing the oriented subbands with a low-pass band:

$$C^O{}_{jk}(x, y) = \frac{\psi_j * \eta_k * L(x, y)}{\phi_{j+2} * L(x, y)}. \tag{2.6}$$

Both of these approaches yield similar results in the decomposition of natural images.

Achromatic contrast sensitivity is generally higher than that of chromatic, especially for high spatial-temporal frequencies. The chromatic CSFs for red-green and blue-yellow stimuli are very similar and the blue-yellow sensitivity is lower. Hence, the full range of colors can only be perceived at low frequencies. As spatial-temporal frequencies increase, blue-yellow sensitivity declines first. At even higher frequencies, red-green sensitivity diminishes as well, and perception becomes achromatic. On the other hand, achromatic sensitivity decreases at low spatial-temporal frequencies, whereas chromatic sensitivity does not.

2.3 Color Perception

In the most general form, light can be described by its spectral power distribution. The human visual system, however, does not process all of the information available in the spectral distribution. The visual system represents colors as a function of the spectral properties of light. There exist lights with different spectral power distributions that cannot be distinguished by a human observer. Thus physically different lights can produce identical color appearance.

Masking and adaptation are two important behaviors in human vision system as they describe interactions between stimuli. Results from masking and adaptation experiments were also the major motivation for developing a multi-channel theory of vision. Masking happens when a stimulus that is visible by itself can not be detected due to the existence of another. Masking is strongest when the interference stimuli have similar characteristics, such as spatial frequencies, orientations, colors, etc. For example, a compressed image looks no difference as the original one because the distortion is masked by the original image that acts as background.

2.4 Multi-Channel Organization

Neurons are tuned to certain types of visual information such as color, frequency and orientation. Data from experiments yielded evidence that these stimulus characteristics are processed in different channels in the human visual system. This empirical evidence motivated the multi-channel theory of human vision [8].

A large number of neurons in the primary visual cortex have receptive fields that resemble Gabor patterns [9]. Hence they can be characterized by a particular combination of spatial frequency, orientation and phase. Serving as an oriented band-pass filter, one such cell thus responds to a certain range of spatial frequencies and orientations. With a sufficient number of appropriately tuned cells, all orientations and frequencies in the sensitivity range of the visual system can be covered. Some cells respond only to oriented stimuli of a certain size. They are sensitive to corners, curvature or sudden breaks in lines.

Temporal mechanisms have been studied as well, but there is less agreement about their characteristics than that of spatial mechanisms. It is now believed that there

is just one low-pass and one band-pass mechanism [10]-[12], which are referred to as the sustained and transient channel, respectively. A small percentage of these cells respond well only when a stimulus moves across their receptive field in a certain direction. These direction-selective cells probably play an important role in motion perception.

2.5 Conclusions

A number of important concepts of human vision system have been introduced in this chapter. The major points can be summarized as follows:

1. While the human visual system is highly adaptive, it is not equally sensitive to all stimuli. There are a number of inherent limitations, such as spatial and temporal frequencies, resolution, contrast, and color.

2. The response of the human visual system depends much more on the local relative contrast than on the absolute luminance, while the local relative contrast is a function of frequency, luminance, color, orientation, etc.

3. Visual information is processed in different channels in the human visual system depending on its characteristics such as color, spatial and temporal frequencies, orientation, phase, direction of motion, etc. These channels are not totally isolated and their interactions with each other play an important role.

These basic concepts will be used to direct our research in this work, in particular in developing the image quality metric in the next chapter, which implicitly used a simplified model based on the most important properties of the human visual system: local relative contrast, resolution limitation, multi-channel.

CHAPTER 3

IMAGE QUALITY ASSESSMENT IN THE WAVELET DOMAIN

3.1 Introduction

To compare the performances of any two image compression methods, both the compression ratios and the qualities of the compressed images have to be considered. An ideal compression system should represent the original image with as small amount of bits as possible while maintaining a good visual quality. In reality, it is always objective in measuring the compression ratio, but highly subjective to judge the quality. Since humans are the end user of images, the natural way to compare the quality of two images is to have them evaluated by human observers. Typically, a group of observers examine a set of images under a controlled environment and assign a numerical score to each of them. Each image's scores are recorded and averaged later as its Mean Opinion Score (MOS) [13] that is by far the most accurate and reliable objective Image Quality Metric (IQM). Unfortunately, MOS is inconvenient and expensive to use.

In [14], ten quality metrics were evaluated against subjective human evaluation. The evaluation was conducted on five different distortion types with variant degrees

of impairments. It is claimed that there still exists difference between machine and human evaluations of image quality, and it is difficult to invent a quality assessment algorithm that is superior in every distortion type. This work is motivated by the need for simple IQMs that are consistent with MOS and suitable for computer implementation. By "consistent", we mean that a metric should perform the same regardless of the distortion types or patterns of the images and be linearly correlated to MOS. That is, it is accurate (giving the same IQM score to images that have the same MOS scores), and increases or decreases monotonically with MOS.

According to its dependence on the original image, an IQM can be classified into three categories:

1. Full-Reference (FR). A Full-Reference metric requires that the original image is available and therefore be used to evaluate the quality of the distorted image. This is the most common category.

2. Reduced-Reference (RR). A Reduced-Reference metric evaluates the quality of the distorted image with only partial knowledge of the original one.

3. No-Reference (NR). A No-Reference metric evaluates the quality of a distorted image without the knowledge of the original one.

This work will focus on Full-Reference IQMs. The most common IQMs are the Mean Squared Error (MSE) family, including MSE, root MSE (RMSE), and Peak Signal to Noise Ratio (PSNR), which are simple pixel error based and their performances are far from satisfactory [15]. Some more sophisticated pixel error based IQMs are also available, such as the method of Damera-Venkata et al. in [16], whose performance, however, is not substantially better than the others [17]. The limitation

of simple pixel error based metrics is also experienced in applications of medical images such as in [18], where the compressed diagnostic breast images with lower PSNR values are preferred by doctors over those with higher PSNR values. That is, the images favored by PSNR do not agree with the judgment of human eyes.

Wang and Bovik proposed a Structural SIMilarity index (SSIM) that models the total distortion of an image block as the combination of three factors: loss of correlation, luminance distortion, and contrast distortion [19]. SSIMs are measured for blocks of an image using a sliding window, and the mean value of the SSIMs (MSSIM) of all the blocks is taken as the overall quality metric of the image. In [20], Shnayderman et al. explored the feasibility of Singular Value Decomposition (SVD) in developing a new IQM that can express the quality of distorted images. An image is first divided into small blocks. The distance between the singular values of the original image block and the singular values of the distorted image block is used to indicate its quality. The overall quality of the distorted image is measured by the absolute value average of differences between these singular value distances and their median. The author claimed that better performance was achieved with smaller block size, which suggested that single pixel based measurement will have the best result. This, in fact, undermined the foundation of their work since singular value decomposition makes no sense for single pixel based measurement. In spite of the differences, these metrics have the same drawback in which they are determined in the spatial domain while compression is performed in the frequency domain, which makes it very difficult to control the visual quality during the compression.

A great deal of effort has been made to develop IQMs that fit the Human Visual System (HVS). While some metrics yield decent results, most are not always consistent with HVS and are sometimes limited to very specific applications. Furthermore, these metrics tend to be complex for implementation. Watson et al. developed a Discrete Cosine Transform (DCT) based video quality metric that incorporates quite a few aspects of human visual sensitivity in [21], and a simple IQM was proposed by Sendashonga and Labeau for both DCT and Discrete Wavelet Transform (DWT) in [22].

In general, compression technologies can be classified into two categories: lossless and lossy. Lossy compression technologies usually first transform the image into the frequency domain, and then quantize/truncate its coefficients. Two most common options of transformation are DCT and DWT, respectively. Compared with DCT, coefficients of DWT are localized in both spatial and frequency domains. That is desirable because HVS functions as a bandpass filter with the localization property [23]. After lossy compression, an image can not be perfectly reconstructed from the quantized coefficients because some data have been truncated or thrown away, which reduces the data size and also generates distortions in the compressed image as a side effect. Distortions can also be introduced by the transformation because of the limited precision of digital computers or the rounding of integer operations, which, however, can be ignored comparing to that caused by quantization. Quantization is a process that has coefficients divided by a numeric value called the quantization step and rounds them to integers to reduce their magnitudes. The original coefficients can not be perfectly recovered from the quantized coefficients because of the rounding error. Quantization, including Scalar Quantization (SQ) and Vector Quantization (VQ)

[24]-[30], plays a very important role in lossy image compression. It is the primary contributor to high compression ratio, and likewise the major source of distortion. In [31], Watson et al. analyzed the DWT quantization errors and developed a quantization algorithm that is aimed to achieve visually lossless compression, but does not have the flexibility to achieve arbitrary visual quality. In [32], Liu et al. developed a quality constrained compression method for JPEG2000 that is optimized for the local profile of so called just-noticeable distortion (JND), which is similar to the distortion model in [31]. In [33], Nadenau et al. came up with a wavelet based color image compression that improved the precision of the contrast sensitive function (CSF), which is complicated and not able to adjust the visual quality.

In this work, we propose a new quality metric called Weighted Normalized Mean Square Error of wavelet subbands (WNMSE), which is defined in terms of the wavelet coefficients and uses the sum of the weighted normalized mean square error of the coefficients in each wavelet subband to assess the quality of a compressed image. This metric is consistent with HVS as well as measures the post-compression quality of an image in real-time because of the simplicity of WNMSE. Taking advantage of WNMSE, we have developed a novel compression algorithm called Quality Constrained Scalar Quantization (QCSQ) that is based on the relationship among the statistic features, quantization steps, and WNMSE value of the image. QCSQ can find the quantization steps for all the subbands efficiently for compressing the image to a desired visual quality measured by WNMSE.

The work is organized as follows. In Section 3.2, we briefly describe the DWT and define the notations that are used in the work. In Section 3.3, our new quality metric

WNMSE is presented, and in Section 3.4, the innovative quality constrained quantization algorithm QCSQ is introduced. Experimental results are given in Section 3.5 to demonstrate the advantages of the new metric and compression methods. The work is concluded by Section 3.6. The detailed algorithm of QCSQ is given in Appendix A.

3.2 2-D Wavelet Transform

The history of wavelet can be traced back to Haar's work in 1909. Starting from the 1980s, contributions to wavelet theory began to boom, such as Goupillaud, Grossmann and Morlet's formulation of what is now known as the Continuous Wavelet Transform (CWT), Strömberg's early work on discrete wavelets, Daubechies' orthogonal wavelets with compact support, Mallat's multiresolution framework, Delprat's time-frequency interpretation of the CWT, and Newland's Harmonic wavelet transform plus many others.

Subband coding, which includes wavelet coding, was first introduced by Croisier et al. for speech coding in 1976 [34]. Ten years later, 2-D subband decomposition was applied to image coding by Woods and O'Neal [35]. With the advent of the wavelet theory, wavelet coding became the dominant subband coding. Figure 3.1 is the diagram of a single-level 2-D wavelet decomposition system, in which four wavelet subbands are generated from the input image and labeled as LL, LH, HL and HH, respectively, where L means low pass filtering and H means high pass. From Figure 3.1, one can see that subband LL is the result of two low pass filtering operations in both the horizontal and vertical directions, while subband LH is the result of a

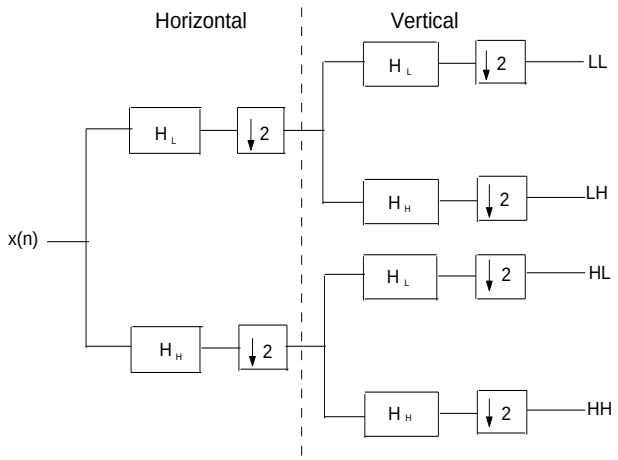


Figure 3.1: The structure of a single-level 2-D subband decomposition system, where H_L represents a low pass filter and H_H represents a high pass filter.

low pass filtering operation in the horizontal direction and a high pass filtering operation in the vertical, respectively; so forth and so on. A balanced multilevel subband decomposition system can be constructed by applying single-level decomposition systems to all the subbands of the previous level. The wavelet transform is the extreme form of an unbalanced subband decomposition because only the subband LL of the previous level is further decomposed.

For convenience, we label subband LL as subband a (average), HL as h (horizontally high pass and vertically low pass), LH as v (vertically high pass and horizontally low pass) and HH as d (both horizontally and vertically high pass). Figure 3.2 is a decomposed image after three levels of 2-D Haar wavelet transform. There are totally ten subbands which can be put into 3 groups according to the levels of transformation: level-1, level-2, and level-3, respectively. After the first transformation, we get four subbands of level-1: a_1, h_1, v_1 and d_1; after applying the second wavelet transformation to a_1, we get four subbands of level-2: a_2, h_2, v_2 and d_2; finally, we get four

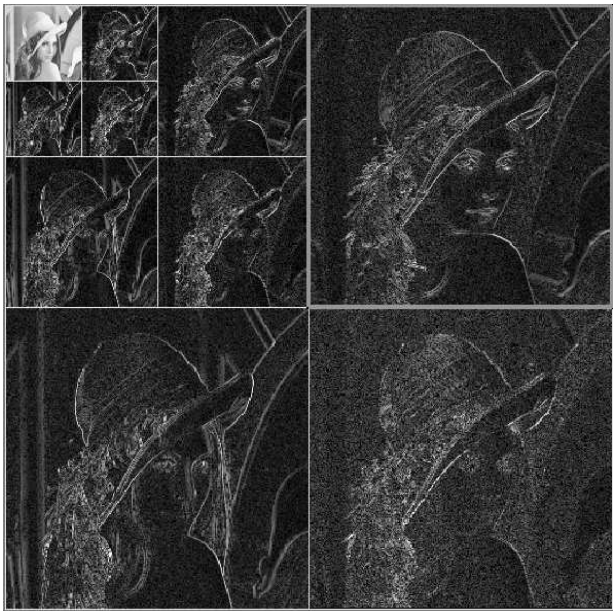


Figure 3.2: The decomposed image of Lenna after three 2-D Haar wavelet transformations

subbands of level-3: a_3, h_3, v_3 and d_3 by applying the last wavelet transformation to a_2. The same operation can continue by applying the 2-D wavelet transform to a_n, $n = 1, 2, 3, ...$, until a_n becomes a single coefficient. However, too many levels of transformation will not contribute to the efficiency of image compression, but only increase the cost of computation.

Besides its level of transformation l, another property of subband b_l is its frequency index f_{b_l}, where b is one of $\{a, h, v, d\}$. A wavelet subband is formed by letting the coefficients passing through a series of filters which includes high pass H_H and low pass H_L, each selectively picking appropriate frequency components. If we let the number of high pass filters that subband b_l passed through be NH_{b_l} and low pass filters NL_{b_l}, we define its frequency index as $f_{b_l} = NL_{b_l} - NH_{b_l}$. In the case above with

$n = 3$, the frequency indexes of the ten subbands $\{d_1, v_1, h_1, d_2, v_2, h_2, d_3, v_3, h_3, a_3\}$ are $\{-2, 0, 0, 0, 2, 2, 2, 4, 4, 6\}$.

The main advantages of using DWT for image coding are:

1. Compared with DCT, the coefficients of DWT are well localized in not only the frequency, but also the spatial domains. This frequency-spatial localization property is highly desired for image compression.

2. DWT decomposes an image into spatially correlated subbands that hold different frequency components of the image. Each subband can be thought as a subset of the image with a different spatial resolution such that the visual quality and the compression ratio of the compressed image can be controlled by adjusting the distortions of different subbands.

3. Images coded by DWT do not have the problem of block artifacts which the DCT approach may suffer [36].

4. Compared with DCT, DWT has lower computation complexity, O(N) instead of O(NlogN) [37].

Xiong et al. claimed that, for still image compression, wavelet transform based coding systems outperform DCT by an order of 1 dB in PSNR [38]. One example of DWT's success is JPEG2000 where 2-D DWT is used instead of DCT.

3.3 The New Quality Assessment Method

Human visual system takes in both frequency and spatial information following a filtering process, and different frequency portions of an image have different contributions to the visual quality. Distortions at different frequencies, even with the

same magnitude, do not have the same impacts to the quality of the compressed image. We define the distortion in the spatial domain as the distance between the pixels of the original image and those of the distorted image, and the distortion in the frequency domain as the distance between the coefficients of the original image after transformation and those of the distorted image after transformation. For distortions in the spatial domain with the *same* magnitude, their corresponding distortions in the frequency domain are *combinations* of distortions of all the subbands. Although the distortion in the frequency domain is related to that in the spatial domain, given the spatial distortion, it is impossible to differentiate the contribution of each subband. An identical distortion index in the spatial domain may attribute to two compressed images which have radically different qualities. Figure 3.3 shows two reconstructed images with the same PSNR, among which the distortion of 3.3(a) is only from the a_1 subband while that of 3.3(b) is from the h_1, v_1 and d_1 subbands. We can see that, the quality of 3.3(a) is worse than that of 3.3(b) even though they have the same amount of distortion in the spatial domain. Since the spatial distortion is not a good indicator of the true quality for human eyes, an image quality metric which is consistent has to be developed in the frequency domain.

The 2-D wavelet transform decomposes an image into subbands that represent different frequency components of the image. Let $x_{b_l,i,j}$ denote a wavelet coefficient before compression and $y_{b_l,i,j}$ the coefficient after compression at position (i, j) in subband b_l. The distortion on this coefficient is $D = |x_{b_l,i,j} - y_{b_l,i,j}|$. In the remaining part of this section, we will analyze how the distortions from different subbands affect the quality of the reconstructed image. For convenience, our analysis is based on the example using Haar wavelet, but the conclusion is applicable to all types of wavelets.

(a) With distortion in only a_1 subband

(b) With distortion in h_1, v_1 and d_1 subbands

Figure 3.3: Two reconstructed images with the same amount of spatial distortions: (a) only has distortion in a_1 subband while (b) has distortions in h_1, v_1 and d_1 subbands. The visual quality of (a) is much worse.

Before we introduce the new quality metric, the following observations are in order.

1. The subbands with higher transformation levels hold more structural or global information, such as shape and luminance, than those with lower transformation levels. So the distortions from the subbands of higher transformation levels degrade the quality of an image more significantly. For example, each coefficient in a level-1 subband comes from four image pixels. If one coefficient has a distortion, it is very likely that those four pixels will all have distortion after reconstructing. Similarly, each coefficient in a level-2 subband comes from sixteen pixels and its distortion will affect those sixteen pixels in the reconstructed image. In a word, any distortion on a coefficient in a level-l subband will generate distortion on each of the 4^l pixels in the reconstructed image, and smaller

distortions in a higher level subband may have more negative impact on the quality of an image than the larger ones in a lower level subband.

2. The subbands of lower frequency (larger frequency indexes) hold more structural or global information than those of higher frequency (smaller frequency index). Since the structural information plays a more important role in maintaining the fidelity of an image, a subband with larger frequency index has more visual impact than that with smaller frequency index. Figure 3.4 shows that the same amount ($NMSE = 10\%$) of distortion produced by subbands with nonidentical frequency indexes has different impact on image quality. Figure 3.4(a) only has distortion in subband a_3 ($f_{a_3} = 6$) while 3.4(b), 3.4(c) and 3.4(d) in h_3 ($f_{h_3} = 4$), v_3 ($f_{v_3} = 4$) and d_3 ($f_{d_3} = 2$), respectively. The quality of 3.4(a) is the worst, 3.4(b) and 3.4(c) next, and 3.4(d) the best.

In light of the above discussion, we believe that a good IQM should be defined in the frequency domain in order to utilize this subband dependent feature. Our new quality metric chooses to use the weighted sum of normalized mean square errors of the coefficients in all the wavelet subbands as the quality metric of an image, which is called the Weighted Normalized Mean Square Error of wavelet subbands (WNMSE):

$$WNMSE_1 = \sqrt{4^{(L-1)} \times 2^{f_{a_L}/2}} \times NMSE_{a_L} + \sum_{b \in \{h,v,d\}} \sum_{l=1}^{L} \sqrt{4^{l-1} \times 2^{f_{b_l}/2}} \times NMSE_{b_l} \tag{3.1}$$

where $\sqrt{4^{l-1} \times 2^{f_{b_l}/2}}$ is the weight factor for subband b_l whose transformation level is l and frequency index f_{b_l}, L is the highest transformation level, $NMSE_{b_l}$ is the Normalized Mean Square Error (NMSE) of subband b_l, and

$$NMSE_{b_l} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{b_l,i,j} - y_{b_l,i,j})^2}{\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{b_l,i,j})^2} \tag{3.2}$$

where m is the number of pixels in the horizontal direction and n vertical. In the case that $\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{b_l,i,j})^2 = 0$, let $NMSE_{b_l} = 0$ if $\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{b_l,i,j} - y_{b_l,i,j})^2 = 0$, and $NMSE_{b_l} = 1$ otherwise. For convenience, we define $WNMSE$ as:

$$WNMSE = 20 \times \log 10\frac{100}{WNMSE_1}. \tag{3.3}$$

In this way, a better quality image will have a higher value of WNMSE which is similar to PSNR and MSSIM.

In this equation, each $NMSE_{b_l}$ is calculated and weighted individually and separately, which reflects the contribution of each subband to the total distortion. *NMSE*, instead of *MSE*, is used because the absolute amount of the distortion is not a good indicator of the contribution of a subband towards the overall quality loss. As discussed above, with the transformation level going up, the number of supporting pixels of a coefficient and its impact to the global structure both increase. By putting 4^{l-1} in the weight factor for subband b_l, its weight goes along with its level. Similarly, by putting $2^{f_{b_l}/2}$ in the weight factor, the impact of frequency is considered accordingly. A subband with higher transformation level and lower frequency will have larger weight. These weights loyally represent the contribution of each wavelet subband to the overall visual quality.

Unlike the conventional quality metrics, WNMSE evaluates the quality of an image in the wavelet domain, which possesses the following two advantages:

1. WNMSE is HVS optimized. Using the weighted contributions of different subbands in the wavelet domain, WNMSE does not simply evaluate the quality of

an image by its total distortion, but treats subbands discriminatingly because different subbands have non-uniform impacts to visual quality. By using different weights, the contribution of each wavelet subband to the overall quality is considered accordingly. In this way, the impacts of distortions to both global structure and local details are more likely to be balanced, which leads to a more objective quality assessment.

2. WNMSE is real-time suitable. By defining WNMSE in the wavelet domain, the quality can be easily assessed during the process of compression. In contrast to those quality metrics in the spatial domain, WNMSE can measure the quality of an image right after quantization without a new computation in the spatial domain. Computation is thus more efficient, especially when iteration is necessary to adjust the quality of the image.

Our research shows that WNMSE is much more consistent with the results of MOS, compared with PSNR and MSSIM. WNMSE is thus a better quality indicator of an image by HVS. In addition, it enables us to link the two operations, quality assessment and quantization during compression, because both of them operate in the frequency domain. With the linkage established, accurate quality constrained compression becomes possible. The experimental results which compare the performance of WNMSE with that of PSNR and MSSIM are provided in Section 3.5.1.

3.4 Quality Constrained Compression

Image compression is usually treated as a bit-rate constrained problem, i.e., compression ratio is on the top of consideration while quality is secondary. Since the

features of images may vary significantly, image qualities can be different for the same bit-rate. Consequently, bit-rate constant compression is not always desired.

We call a compression method which prioritizes the quality *quality constrained compression*. Unfortunately, quality constrained compression has been difficult because of the following two reasons:

1. Quality assessment, such as PSNR and MSSIM, and image compression, such as DCT or DWT based, are pursued in the spatial and frequency domains, respectively, and there is no direct and simple link between them.

2. The reliability of current IQMs still have to be improved to satisfy the need of the quality assessment.

These two problems can be solved by using the new index WNMSE. From Equation (3.1), the WNMSE of a compressed image can be controlled if the distortion of each wavelet subband can be manipulated. This could be done through a brutal-force searching method, but an applicable solution has to be more efficient. Ideally, we want to be able to predict the distortion caused by a given quantization step. This appears to be a challenging task because it requires a highly accurate statistical description of the subband. Many efforts have been made to develop statistic models of wavelet coefficients and employ them in image compression. Unfortunately, they are often inaccurate in the modeling, and not easy to use [39]-[42]. From the discussion of the previous section, one can see that choosing of the step for a particular subband must be related to its contribution to the quality of the image. Large contributors should have less distortions, i.e., smaller steps. The question is who are the large contributors? We propose to predict the contribution of a subband by a set of

features, and use these features to select the initial step and subsequently tune it to reach the desired quality. These features are *transformation level*, *frequency index*, *energy level*, *standard deviation*, and *complexity*, respectively. While the definitions of the transformation level and frequency characteristic have been described earlier and that of the standard deviation is trivial, the other two features are defined below.

1. Since the energy of a subband is calculated as the sum of the squares of each coefficient, it depends only on the absolute magnitude of each coefficient. So we use the absolute mean value m_{b_l} to represent the *energy level* of subband b_l.

$$m_{b_l} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} |x_{b_l,i,j}|}{n \times m} \tag{3.4}$$

   where $x_{b_l,i,j}$ is a coefficient of subband b_l at position (i, j), and m and n is the dimensions of subband b_l.

2. At the first glance, the standard deviation σ_{b_l} of the wavelet coefficients in subband b_l is the only parameter needed to represent the complexity of the subband.

$$\sigma_{b_l} = \sqrt{\frac{\sum_{i=1}^{n} \sum_{j=1}^{m} (x_{b_l,i,j} - \overline{x_{b_l}})^2}{n \times m - 1}} \tag{3.5}$$

   where

$$\overline{x_{b_l}} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} x_{b_l,i,j}}{n \times m}. \tag{3.6}$$

   It is not enough because the energy levels of subbands could be different. For example, two subbands with identical standard deviations of 10, may have absolute means of 50 and 5, respectively. In this situation, the two subbands do not have the same complexity level. So we use the "relative" standard deviation $vm_{b_l} = \sigma_{b_l}/m_{b_l}$ (std/mean) to represent the *complexity* of subband b_l.

It is well known that the subbands of the wavelet transformation are projections of the original image to various resolutions, and their energy levels and complexities are related to each other. We can simply use the energy level and complexity of subband a_1, i.e., m_{a_1} and vm_{a_1}, to uniformly represent those of all the subbands.

The impact to the image quality by a particular step is affected by the five features just mentioned. It is not possible to deduct a quantitative relationship between the step and the features for a desired image quality, but it is not difficult to understand the qualitative relationship between the two. Based on these observations, we introduce the following equation for defining the quantization step of subband b_l:

$$s_{b_l} = C_l \cdot V_{b_l} \tag{3.7}$$

where C_l is a variable that is only dependent on the transformation level l, and V_{b_l} is a variable whose value is derived from a function of σ_{b_l} while the function itself is determined by the other four features of subband b_l. Accordingly, to get a high compression ratio while satisfying a quality constrain, C_l and V_{b_l} can be determined using the following rules:

1. V_{b_l} should increase as m_{a_1} increases.
2. V_{b_l} should increase as vm_{a_1} increases.
3. V_{b_l} should decrease as f_{b_l} increases.
4. C_l should decrease as the transformation level increases.
5. V_{b_l} should be proportional to σ_{b_l}.
6. The quality and compression ratio of a compressed image can be tuned by adjusting its quantization steps to achieve an optimal result.

Using the rules just defined, a process has been found to search for quantization steps for compressing an image. This process is called Quality Constrained Scalar Quantization (QCSQ) which takes two steps: first, find the initial set of steps which is

nearly optimal in the compression ratio with a uniform quality metric WNMSE ≈ 28, where 28 is chosen as the lower bound of an acceptable visual quality. Secondly, tune the initial steps to increase the quality of an image to a desired value.

3.4.1 Find the Initial Set of steps

When calculating the WNMSE indexes, we multiply the NMSE of a subband b_l by $\sqrt{4^{l-1} \times 2^{f_{b_l}/2}}$, where 4^{l-1} is dependent on its transformation level and $2^{f_{b_l}/2}$ is dependent on its frequency. Here the dependence of step on the transformation level is reflected by defining the variable $C_l = 4^{(L-l)}$. The impact of the frequency index is reflected by multiplying a factor $2^{-f_{b_l}/2}$ when calculating V_{b_l}. The detailed implementation of this algorithm is in Appendix A.1.

3.4.2 Tune the Initial Set of steps

An image quantized by its initial set of steps only achieves the lower bound of visual quality. By further tuning its steps, one can improve the quality of the image to a desired level. Figure 3.5 shows how the variations of the steps of different subbands alternate the quality of images. We use two empirical parameters to evaluate the efficiency of the step tuning of a subband: quality gain and optimality. The quality gain of subband b_l is reduced from the quality improvements of images with different features by reducing the step of subband b_l by half. By optimality, we mean the ratio between the quality increment (ΔQ) and the compression ratio decrement (ΔR): (ΔQ) / (ΔR). Since we want to maintain as high a compression ratio as possible when increasing the quality, ΔR should be as small as possible; therefore, the higher the ratio (ΔQ) / (ΔR) is, the higher the optimality level is.

Tuning Order	1	2	3	4	5	6	7	8
Subband	h_1	d_3	v_1	d_2	h_3	v_3	v_2	h_2
Quality Gain (Haar)	0.58	0.47	0.58	0.50	0.14	0.13	0.18	0.18
Quality Gain (5/3)	0.57	0.49	0.58	0.52	0.13	0.13	0.18	0.18
Quality Gain (9/7)	0.57	0.49	0.56	0.53	0.13	0.13	0.18	0.18
Quality Gain (DB4)	0.51	0.49	0.54	0.55	0.13	0.13	0.17	0.18
Quality Gain (4/4)	0.50	0.49	0.53	0.54	0.13	0.13	0.19	0.18
Quality Gain (6/2)	0.58	0.47	0.59	0.50	0.13	0.13	0.18	0.19

Table 3.1: Order of the subbands for fine-tuning and the predicted quality gains by reducing their steps by half.

Figure 3.5(a) shows the normalized optimality of each subband, which is sorted in the ascending order, and Figure 3.5(b) shows the magnitude and variance of the quality gain of each subband. To achieve accuracy, efficiency, and high compression ratio, only those subbands that have low quality gain variances, high quality gains, and high optimality values are used for quality tuning. Since the initial steps give the lower bound of the visual quality of an image, only the tuning for quality increase is considered. Combining the results of Figure 3.5(a) and Figure 3.5(b), the following rules of fine-tuning are obtained:

1. If there is more than one choice satisfying the quality requirement, choose the one which has the maximum compression ratio.

2. Tune the steps of the subbands with higher optimality first.

3. Tune only subbands whose quality gains are more than 0.1.

4. Tune only subbands whose variance of quality gains is less than 0.66.

5. Reduce the step by half when tuning it (because of the binary property of digital data).

The resulting order of tuning and the expected quality gain for each fine-tuning are listed in Table 3.1, where the values shown are the average of 31 different images. We can see that the tuning orders are identical for all the wavelets and the quality gains show little difference. For a specific image, the quality gain may be slightly different, but the order of tuning is universally true.

Since WNMSE is defined in the wavelet domain, we can easily measure it after quantizing an image with the initial steps. Let the initial WNMSE be Q_0 and the objective WNMSE be Q, the difference is $\Delta Q = Q$ - Q_0. To increase the quality metric by ΔQ, we should tune the steps following the rules above. The detailed implementation of this algorithm is in Appendix A.2.

3.5 Experimental Results

In this section, we first compare the quality assessment performance of WNMSE with that of PSNR and MSSIM, and then use an example to show how to achieve quality constrained compression with QCSQ. The Haar, DB4, 5/3 and 9/7 wavelets are used in our experiments to show the generalization of the algorithm.

3.5.1 Compare the Performance of WNMSE with PSNR and MSSIM

The performance of WNMSE is compared with that of PSNR and MSSIM by applying them to two sets of images, which includes twenty four and twenty five degraded images, respectively. The impairments of the degraded images are either from

Score	Description
5.0	Perfect. The distortion is imperceptible
4.0	Good. The distortion is perceptible, but not annoying
3.0	Fair. The distortion is slightly annoying
2.0	Bad. The distortion is annoying
1.0	Very bad. The distortion is very annoying
0.0	Unidentifiable. The image is totally ruined

Table 3.2: The reference table of ranking scores.

compression with JPEG or JPEG 2000, or various amounts of additive noise, including Gaussian, Speckle and Salt-pepper. These images are independently evaluated by 12 persons who come from different backgrounds. Three of them are considered as experts since they work in the image processing field, and the others are non-experts. To evaluate the qualities of these images, a person gave each degraded image a score using Table 3.2 as a reference. Each score is from 0.0 to 5.0 including a decimal fraction of one digit. The average score of an image is taken as the Mean Opinion Score (MOS) of it. By comparing the MOS indexes given by WNMSE, PSNR, and MSSIM, the accuracy of WNMSE is higher than both PSNR and MSSIM. Among the images we used, Lenna and Peppers are the mostly used ones. Without loss of generality, they are chosen as two visual examples to prove that the performance of WNMSE is better.

We also used four popular criteria to evaluate the accuracy of the quality metrics. Among them, the first three are the standard criteria used by the Video Quality Expert Group (VQEG) [43], and the fourth is straight "Sum of Squared Errors". In the following definitions, "X" can be "PSNR", "MSSIM" or "WNMSE", X_i is the

normalized "X" and MOS_i the normalized MOS of the ith image, and n is the number of images.

1. Pearson Linear Correlation Coefficient ($PLCC$) is used to evaluate the *accuracy* of an IQM. The $PLCC$ of "X" with regard to MOS is

$$PLCC_X = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(MOS_i - \overline{MOS})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(MOS_i - \overline{MOS})^2}}. \tag{3.8}$$

   The larger the $PLCC_X$ is, the more accurate X will be with regard to MOS.

2. Spearman Rank Order Correlation Coefficient ($SROCC$) is used to evaluate the *monotonicity* of an IQM. The $SROCC$ of "X" with regard to MOS is

$$SROCC_X = 1 - \frac{6\sum_{i=1}^{n}(d_i)^2}{n^3 - n} \tag{3.9}$$

   where d_i is the difference between each rank of corresponding values of X and MOS. The larger the $SROCC_X$ is, the better monotonicity X will have with regard to MOS.

3. Outlier Ratio (OR) is used to evaluate the *consistence* of an IQM. The OR of "X" OR_X with regard to MOS is defined as the number of outliers divided by n, where twice of the standard error of MOS was used as the threshold for defining outliers. From the definition we can see that the smaller the OR_X is, the more consistent X will be with regard to MOS.

4. Sum of Squared Errors (SSE). We also use SSE of an index with regard to MOS to measure the overall performance of it. The values of every index are normalized so that they all have the same range between [0, 1].

$$SSE_X = \sum_{i=1}^{n}(X_i - MOS_i)^2. \tag{3.10}$$

	Evaluation Metrics	WNMSE (Haar)	WNMSE (DB4)	WNMSE (5/3)	WNMSE (9/7)	PSNR	MSSIM
Image series A n=24	SSE	0.606	0.750	0.873	0.791	1.347	1.453
	PLCC	0.7211	0.7138	0.6580	0.6922	0.5306	0.4877
	SROCC	0.7457	0.7439	0.6896	0.7300	0.5809	0.5326
	OR	0.0420	0.1250	0.1250	0.1250	0.1250	0.1250
Image series B n = 25	SSE	0.549	0.589	0.693	0.615	0.842	1.089
	PLCC	0.7814	0.7679	0.7304	0.7612	0.6558	0.5651
	SROCC	0.8485	0.8354	0.7632	0.8285	0.7573	0.6562
	OR	0.0800	0.0800	0.0800	0.0800	0.0800	0.0800

Table 3.3: Overall performances of WNMSE, PSNR and MSSIM measured by their Sum of Squared Errors (SSE) with regard to MOS.

The smaller the SSE_X is, the better X will perform with regard to MOS.

The evaluation results of the quality metrics are listed in Table 3.3. WNMSE implemented with four DWTs are compared with PSNR and MSSIM. Looking at the table, we can see that all the WNMSEs outperform PSNR and MSSIM in every aspect while the Haar WNMSE is the best.

Figures 3.6 and 3.7 show how WNMSE outperforms both PSNR and MSSIM. Image (a) is the original image and the other three are degraded by Gaussian noise, Salt-Pepper noise, and JPEG 2000 compression, respectively, which are listed in the descending order of their MOS values. The measured quality metrics are listed in Table 3.4 and Table 3.5. From the measured quality metrics, we can see that the WNMSE indexes are in the same order as those of MOS, while PSNR and MSSIM give the reverse results. This proves that WNMSE functions more like human eyes.

Image	WNMSE Haar	WNMSE DB4	WNMSE 5/3	WNMSE 9/7	PSNR	MSSIM	MOS
b)	19.16	16.99	14.48	15.87	23.06	0.496	3.26
c)	17.28	15.07	12.54	13.93	23.10	0.498	3.07
d)	13.08	10.80	10.03	10.34	24.37	0.605	0.85

Table 3.4: Quality indexes of the image group of Lenna

Image	WNMSE Haar	WNMSE DB4	WNMSE 5/3	WNMSE 9/7	PSNR	MSSIM	MOS
b)	18.90	16.45	13.76	15.24	23.04	0.542	3.25
c)	18.60	16.18	13.16	14.80	24.01	0.601	3.24
d)	17.64	15.16	12.60	14.20	24.35	0.691	1.02

Table 3.5: Quality indexes of the image group of Peppers

3.5.2 QCSQ Examples

In this example, we use 9/7 wavelet which has been used in the JPEG 2000 standard for lossy compression. We first apply three levels of 2-D 9/7 wavelet transform to an image, and then use QCSQ to determine the quantization steps for all its wavelet subbands. After quantization, we use Zig-zag sorting followed by Stack-run [44] to code the compressed image. Entropy coding, such as Huffman coding or Arithmetic coding, has no impact on the quality of images, and was thus not applied in our experiments.

Six images are used in the experiment, where the desired quality index is $Q = 30$ in WNMSE with an acceptable error of 0.3. So the final quality metrics of the six images should be between 29.7 and 30.3 in WNMSE.

1. Find the initial steps and compute the initial quality metrics. According to our algorithm, an image quantized by its initial steps should have an initial quality index $Q_0 = 28$ measured in WNMSE. The results are listed in Table 3.6, from which we can see that the initial quality indexes of all the other five images are distributed closely around 28.0 except Mige171 which has a WNMSE value of 28.31.

2. Tune the steps. We know that $\Delta Q = Q - Q_0 \approx 30 - 28 = 2$ for the other five images and the sum of the quality gains of the first four most optimal subbands: h_1, d_3, v_1 and d_2 (Table 3.1), are 0.57 + 0.49 + 0.56 + 0.53 = 2.15. So we first reduce the steps of these four subbands by half for the five images. As for Mige171 whose $\Delta Q = 1.69$, we only need to reduce the steps of the first three subbands: h_1, d_3 and v_1 which will give a quality gain of 1.62. The resulting quality metrics and compression ratio are listed in Table 3.7. We can see that the WNMSEs of the five images (Lenna, Lethal, Tree, Mige171 and Building) already fall into the desired range. For the Peppers image, the WNMSE is a little too high, which will cause unnecessary loss in compression ratio. If we recover the steps of subband d_2 to the initial setting, its predicted quality metric is $30.70 - 0.53 = 30.17$ that is in the desired range. The measured quality metric after tuning is 30.14 that is only slightly different from the predicted value. Table 3.8 lists the final results, and Figure 3.8 shows the compressed images.

The experimental results have shown that the desired quality metric is achieved with no or only one iteration, and the resulting compression ratios are near optimal.

Image	s_{a_3}	s_{h_3}	s_{v_3}	s_{d_3}	s_{h_2}	s_{v_2}	s_{d_2}	s_{h_1}	s_{v_1}	s_{d_1}	WNMSE	Comp. Ratio
Lenna	42	8	15	16	13	22	22	80	128	128	28.05	32.09
Lethal	45	16	20	21	24	32	30	176	144	128	27.97	34.07
Peppers	46	16	21	21	27	31	30	176	192	224	28.03	26.81
Tree	48	13	10	14	22	18	28	160	112	128	28.08	32.08
Mige171	63	24	18	19	35	30	32	192	176	128	28.31	38.64
Building	46	22	14	17	39	29	32	256	192	160	27.86	29.22

Table 3.6: Initial results before fine-tuning

Image	s_{a_3}	s_{h_3}	s_{v_3}	s_{d_3}	s_{h_2}	s_{v_2}	s_{d_2}	s_{h_1}	s_{v_1}	s_{d_1}	WNMSE	Comp. Ratio
Lenna	42	8	15	8	13	22	11	40	64	128	29.97	27.02
Lethal	45	16	20	11	24	32	15	88	72	128	30.17	29.19
Peppers	46	16	21	11	27	31	15	88	96	224	30.70	22.26
Tree	48	13	10	7	22	18	14	80	56	128	30.21	27.33
Mige171	63	24	18	10	35	30	32	96	88	128	30.24	34.52
Building	46	22	14	9	39	29	16	128	96	160	29.83	25.11

Table 3.7: Intermediate results of fine-tuning

Image	s_{a_3}	s_{h_3}	s_{v_3}	s_{d_3}	s_{h_2}	s_{v_2}	s_{d_2}	s_{h_1}	s_{v_1}	s_{d_1}	WNMSE	Comp. Ratio
Lenna	42	8	15	8	13	22	11	40	64	128	29.97	27.02
Lethal	45	16	20	11	24	32	15	88	72	128	30.17	29.19
Peppers	46	16	21	11	27	31	30	88	96	224	30.14	23.70
Tree	48	13	10	7	22	18	14	80	56	128	30.21	27.33
Mige171	63	24	18	10	35	30	32	96	88	128	30.24	34.52
Building	46	22	14	9	39	29	16	128	96	160	29.83	25.11

Table 3.8: Final results of fine-tuning

3.6 Conclusions

In this work, we have proposed a new quality metric WNMSE and an innovative quantization algorithm QCSQ. WNMSE uses the weighted sum of the normalized mean square errors of wavelet coefficients to assess the quality of an image. According to the concepts of HVS, the weight for each subband is chosen to reflect its perceptual impact on the image, which measures the distortions in the global structure and local details of an image in a more balanced way automatically. Because WNMSE is defined in the wavelet domain, it can be calculated in the middle of compression without reconstructing the image. Furthermore, it facilitates the link between the quantization steps and the quality metric. Our experiments show that WNMSE has better performance than both the legacy PSNR and the well referenced new IQM SSIM.

The features of a subband can be represented by its transformation level, frequency, energy, standard deviation, and complexity, which alternate the effect of the quantization step to the WNMSE of a compressed image. Based on the analysis of the relationship among the subband features, steps, and WNMSE values, we have invented a quality constrained compression algorithm QCSQ which can identify the quantization step for every subband of an image. With these steps, the image can be compressed to a desired visual quality measured by WNMSE.

This work shows that, by developing the quality metric and the quantization algorithm in the same wavelet domain, we have made the quality constrained image compression possible, while pushing the compression ratio as high as possible.

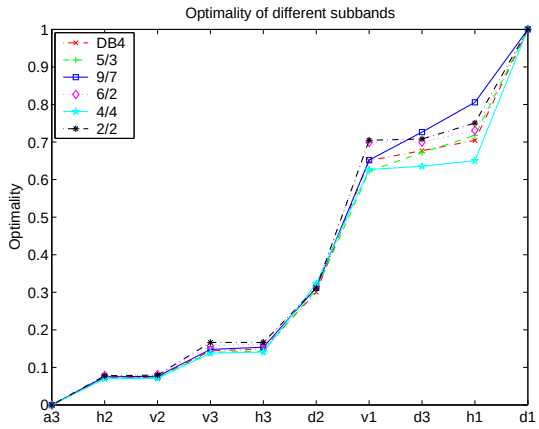
(a) distortion in a_3 subband, $f_{a_3} = 6$

(b) distortion in h_3 subband, $f_{h_3} = 4$

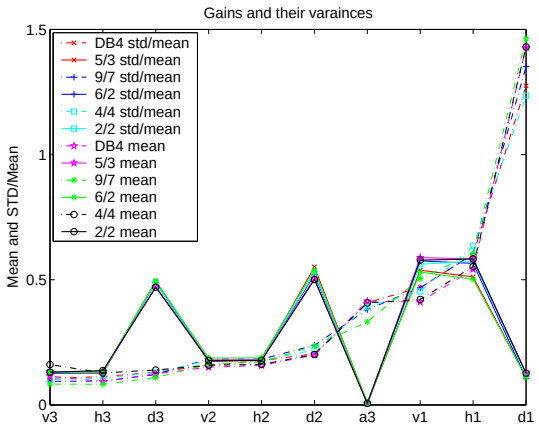
(c) distortion in v_3 subband, $f_{v_3} = 4$

(d) distortion in d_3 subband, $f_{d_3} = 2$

Figure 3.4: Four reconstructed images with the same amount ($NMSE = 10\%$) of frequency distortions: (a) only has distortion in subband a_3 while (b), (c) and (d) in h_3, v_3 and d_3 respectively. The quality of (a) is the worst, (b) and (c) next, and (d) the best.



(a) The optimality level of different subbands when reducing their steps by half to improve the quality of the image. Sorted as the ascending order of their optimality levels.



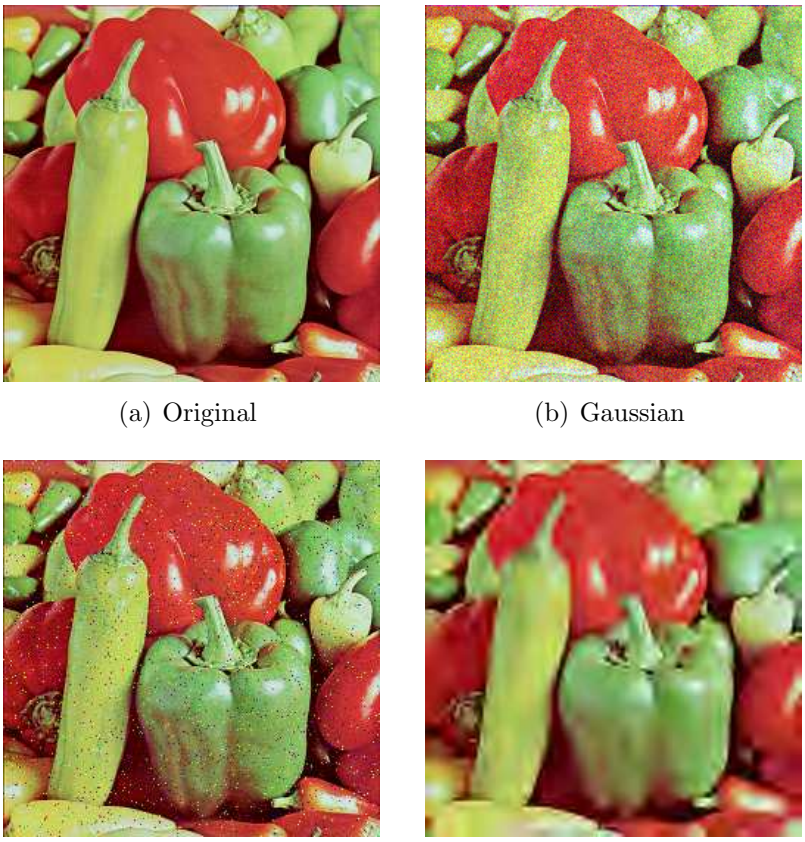
(b) The quality gains of different subbands when reducing their steps by half to improve the quality of the image. Sorted as the ascending order of their normalized standard deviations.

Figure 3.5: When reducing the step, each subband has different quality gains and different optimality levels. Some of them have higher optimality levels and lower invariance of quality gains, which can be used to adjust the quality of the compressed image.

(a) Original (b) Gaussian noise

(c) Salt-Pepper noise (d) JPEG 2000 compression

Figure 3.6: WNMSE outperforms both PSNR and MSSIM for Lenna images: WNMSE and MOS indexes are in the same order as (b) to (d) in the quality measure, but PSNR and MSSIM give the reverse order as (d) to (b).



(a) Original (b) Gaussian

(c) Salt-Pepper noise (d) JPEG 2000 compression

Figure 3.7: WNMSE outperforms both PSNR and MSSIM for Peppers images: WNMSE and MOS indexes are in the same order as (b) to (d) in the quality measure, but PSNR and MSSIM give the reverse order as (d) to (b).

(a) lenna: WNMSE = 29.97.

(b) Tree: WNMSE = 30.17.

(c) Mige171: WNMSE = 30.14.

(d) Building: WNMSE = 30.21.

(e) peppers: WNMSE = 30.24.

(f) lethalweapon: WNMSE = 29.83.

Figure 3.8: Compressed images that are fine-tuned to WNMSE = 30.00.

CHAPTER 4

SUB-GROUPING TRANSFORMATION

4.1 Introduction

Video is basically a three-dimension (3-D) matrix of color pixels. Two dimensions serve as spatial (horizontal and vertical) directions of the moving pictures, and one dimension represents the time domain. A frame is a set of all spatial pixels that correspond to a single point in time, i.e., it is the same as a still picture. Video data contains spatial and temporal redundancy, i.e., the similarities within a frame (spatial redundancy) and between frames (temporal redundancy). To reduce the spatial redundancy, the spatial encoding or intra-frame compression is performed on the current frame to take advantage of the intra-frame correlation, and the limitations of human eyes, such as color, resolution, luminance and contrast. Intra-frame compression is effectively image compression. To reduce temporal redundancy, the temporal coding or inter-frame compression is applied by exploiting the inter-frame correlation. Inter-frame compression uses earlier or later frames (reference frames) in a video sequence to compress the current frame.

Inter-frame compression is a powerful technique for compressing video. The most commonly used method is the block-based Motion-Estimation/Compensation

(ME/C) that works by estimating a frame in a video sequence from its reference frame/frames. If the current frame contains a block that is "similar" enough to a block in the reference frame, the system simply calculates a motion vector that links the block in the current frame to the one in the reference frame so that the decoder can construct that block from the reference one and create the predicted frame. Only the difference between the predicted frame and the current frame needs to be coded by intra-frame compression.

With the release of the JPEG2000, the Discrete Wavelet Transform (DWT) has become the dominant transform in the field of image compression. Discrete Cosine Transform (DCT), still widely used in the older standards, is fading away from the research on image compression. As for video compression, the dominant standards in the market are still DCT-based. Today, nearly all video compression methods in common use (e.g., those in standards approved by the ITU-T or ISO) apply a DCT for spatial redundancy reduction. DWT is typically not used in practical products (except for the use of wavelet coding as still-image coders without motion compensation), but has been the hot subject of research. The video compression algorithms using DWT can be classified into three categories:

1. 2-D wavelet decomposition plus motion-estimation/compensation (ME/C). Coding systems in this category, such as those in [45], [46], [47] and [48], have the similar structure as that of the DCT-based video compression. Block-based ME/C is used to reduce the temporal redundancy.

2. 3-D extensions of 2-D wavelet based image compression algorithms. Coding systems in this category, such as those in [49], [50] and [51], extend the successful

wavelet-based image coding algorithms to the field of video coding. Motion-compensation can also be integrated with algorithms in this category, where the temporal direction wavelet transform is applied along the motion vectors or trajectories.

3. Motion-compensated 3-D wavelet decomposition. This is typically done by applying the temporal wavelet decomposition on the motion-compensated frames instead of the original frames [52], [53], [54].

The DWT decomposes a whole frame into subbands that each contains a lower resolution projection of this frame at different frequency. Thus the encoder knows not only the frequency of a coefficient but also where it is located in the frame. For image compression, this is an advantage over the block-based DCT since it allows the encoder to assign bits to each frequency component of the frame. It is efficient in reducing the spatial redundancy and can better identify which data is more relevant to human perception. Things are more complicated for video compression where temporal (inter-frame) encoding plays an essential role, especially when the compression ratio is very high. The reason is that the spatial decomposition interferes the inter-frame correlation between frames, thus limiting the efficiency and accuracy of motion-compensation which is critical for successful reducing of temporal redundancy.

For video compression, the most important concern may be how to reduce the temporal redundancy without introducing too much temporal distortion. Currently, using motion vectors is the most common choice to address this issue in both DCT and DWT-based video compression systems. In these DCT or 2-D DWT-based video compression systems, motion vectors are used to predict blocks in the current frame from blocks in the reference frames and construct a predicted frame. The difference

between the original and predicted frames is called error frame or residual. In 3-D DWT-based systems, there is no error frame. Instead, video frames are put into groups of frames and the temporal wavelet filtering is applied along the motion vectors or trajectories of each group of frames to compact the energy into the temporal low frequency subbands. In both of the above approaches, motion vectors must be encoded with lossless methods. For applications requiring very low bit rates, the overhead of coding motion vectors is not desired.

One problem for very low bit rate compression is the *ghost effect*, that is, objects only existing in certain frames appear in other frames in the reconstructed video. This is because of the over-reduction of high frequency temporal information. While the loss of high frequency spatial information blurs the frames, the loss of high frequency temporal information results in the averaging of frames along temporal direction, which causes the energy leakage among frames and is the direct reason of ghost effect. If small quantization steps are used to avoid the over-reduction, not only the compression ratio is limited, but also the background noise will survive the quantization and leads to the fluctuating background in the reconstructed video.

To solve the contravention, we propose an innovative inter-frame compression scheme: Sub-Grouping Transformation (SGT), which can achieve high compression ratio and reduce the ghost effect at the same time. The chapter is organized as follows: Section 4.2 gives an overview of the motion-estimation/compensation technology; Section 4.3 describes the proposed sub-grouping transform algorithm; Section 4.4 compares the proposed algorithms with others using experimental results; Section 4.5 will summarize this chapter.

4.2 Motion-Estimation/Compensation (ME/C)

A video sequence consists of a number of pictures - usually called frames. Subsequent frames are very similar, thus containing a lot of temporal redundancy. Removing this redundancy helps achieve the goal of better compression ratios.

A first approach would be to simply subtract a reference frame from a given frame. The difference is then called residual and usually contains less energy (or information) than the original frame. The residual can be encoded with less bits for the same quality. The decoder can reconstruct the original frame by adding the reference frame to the residual again.

Motion-Estimation/Compensation (ME/C) is a more sophisticated approach, which constructs a predicted frame with blocks in the reference frame. The locations of a block in the reference frame and the predicted frame are usually different. The displacement or motion is described by some parameters called motion vectors. Motion vectors have to be losslessly encoded in the bit-stream. The predicted frame will then be subtracted from the original frame to get the residual frame. This generates residual frames with much less energy than those by the previous simple approach. However, the bits consumed by coding the motion vectors become overhead.

ME/C requires videos to be processed in groups of frames (GOF). The first frame that is encoded without motion compensation (just like a still image) is called I-frame. The frames that are predicted from the I-frame or P-frame before it are called P-frames. Frames can also be predicted from future frames. The future frames thus need to be encoded before the predicted frames and the encoding order does not necessarily match the real frame order. Such a predicted frame is usually predicted

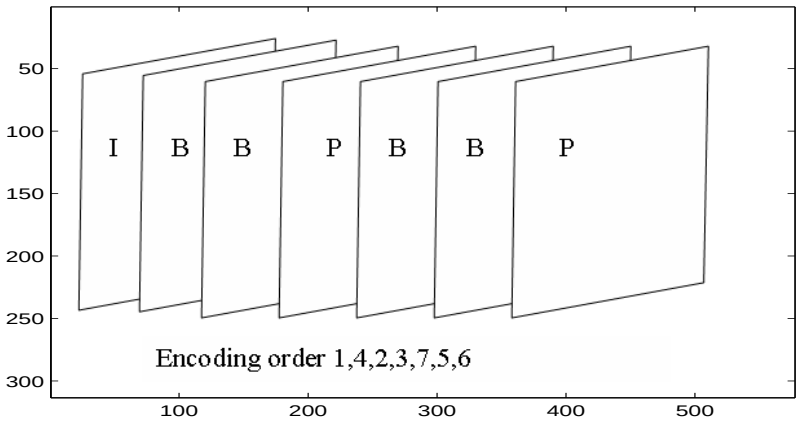


Figure 4.1: Inter-Frame coding

from two directions, i.e. from the I- or P-frame that precedes or follows it. These bi-directionally predicted frames are called B-frames. For example, a coding scheme of a group of frames could be ***IBBPBBP*** and the encoding order is {***1,4,2,3,7,5,6***} (Figure 4.1). Frame 1 is the I-frame that is first encoded independently by intra-frame compression. Frame 4 is a P-frame that uses frame 1 as the reference frame, so it is encoded secondly. Frame 2 and 3 are B-frames that are dependent on both frame 1 and 4, so they are encoded after frame 1 and 4. Frame 7 is another P-frame that only depends on frame 1, so it is encoded next. Frame 5 and 6 are B-frames that depend on both frame 1 and 7, so they are encoded last.

4.2.1 Motion Compensation

Motion compensation includes the global motion compensation and local block-based motion compensation.

Global motion compensation

In global motion compensation, the motion model basically reflects camera motions. It works best for still scenes without moving objects. There are several advantages of global motion compensation:

- It models precisely the major part of motion usually found in video sequences with just a few parameters.
- It does not partition the frames. This avoids artifacts at partition borders.
- A straight line in the temporal direction of pixels with the same spatial position in the frame corresponds to a continuously point in the real scene. Other motion compensation schemes introduce discontinuities in the temporal direction.

However, moving objects are not sufficiently represented by global motion compensation. Thus, other local methods are preferable.

Block-based motion compensation

In block-based motion compensation, the frames are partitioned in blocks of pixels (e.g. macro-blocks of 16×16 pixels in MPEG-2). Each block is predicted from a block of equal size in the reference frame. The blocks are not transformed in any way apart from being shifted to the position of the predicted block. This shift is represented by a motion vector.

The motion vectors are the parameters of this motion model and have to be encoded into the bit-stream. As the motion vectors are not always independent, for example, two neighboring blocks belong to the same moving object, they are usually encoded differentially to save bit-rate. This means that the difference of the motion vector and the neighboring motion vector(s) should be encoded before it is encoded. The result of this process is mathematically equivalent to a global motion compensation capable of panning. An entropy codec can exploit the resulting statistical distribution of the motion vectors. It is possible to shift blocks by non-integer vectors, which is called sub-pixel precision motion compensation. This is done by interpolating the pixel's values. The computational expense of sub-pixel precision is much higher due to the interpolation required.

Block-based motion compensation divides the current frame into non-overlapping blocks, and the motion compensation vector tells where those blocks *come from*. A common misconception is that the previous frame is divided into non-overlapping blocks, and the motion compensation vectors tell where those blocks *move to*. In fact, the source blocks typically overlap in the reference frames. Some video compression algorithms assemble the current frame out of blocks of several previously-transmitted frames.

The main disadvantage of block-based motion compensation is that it introduces discontinuities at the edges of blocks, so called blocking artifacts. These artifacts appear in the form of sharp horizontal and vertical edges which are easily spotted by the human eye.

Variable block-based motion compensation

Variable block-based motion compensation is the use of block-based motion compensation with the ability for the encoder to dynamically select the size of the blocks. When coding video, the use of larger blocks can reduce the number of bits needed to represent the motion vectors, while the use of smaller blocks can result in less amount of prediction residual information to encode. Older designs such as H.261 and MPEG-1 video typically use a fixed block size, while newer ones such as H.263, MPEG-4 Part 2, H.264/MPEG-4 AVC, and VC-1 give the encoder the ability to dynamically choose what block size will be used to perform the motion estimation and compensation.

Overlapped block-based motion compensation

Overlapped block-based motion compensation is a good solution to improve the compression quality because it not only increases prediction accuracy but also avoids blocking artifacts. When using the overlapped block-based motion compensation, blocks are typically twice as big in each dimension and overlap quadrant-wise with all 8 neighboring blocks. Thus, each pixel belongs to 4 blocks. In such a scheme, there are four predictions for each pixel which are summed up to a weighted mean. For this purpose, blocks are associated with a window function that has the property that the sum of four overlapped windows coefficients is equal to 1 everywhere.

4.2.2 Motion-Estimation

Motion-estimation is the process of finding optimal or near-optimal motion vectors. The amount of prediction error for a block is often measured using the mean

squared error or sum-of-absolute-differences between the predicted and actual pixel values over all pixels of the motion-compensated region.

There are two mainstream techniques of motion-estimation: pixel-recursive algorithm [56] and block-matching algorithm. The pixel-recursive algorithms are iterative refining of motion-estimation for individual pixels by gradient methods. The block-matching algorithms assume that all the pixels within a block has the same motion activity. The block-matching algorithms estimate motion on the basis of rectangular blocks and produce one motion vector for each block. In a typical block-matching algorithm, each frame is divided into blocks, each of which consists of luminance and chrominance blocks. Usually, for coding efficiency, motion-estimation is performed only on the luminance block. Each luminance block in the present frame is matched against candidate blocks in a search area on the reference frame. The best (lowest distortion) candidate block is found and its displacement (motion vector) is recorded. In a typical inter-frame coder, the input frame is subtracted from its predicted version from the reference frame. Consequently the motion vector and the resulting residual can be transmitted instead of the original block. Thus inter-frame temporal redundancy is removed and data compression is achieved. At the receiver end, the decoder builds the residual frame from the received data and adds it to the reconstructed predicted frame. The better the prediction is, the less energy the residual frame has and hence lower the transmission bit rate.

To find optimal motion vectors, one basically has to calculate the block prediction error for each motion vector within a certain search range and pick the one that has the best compromise between the amount of error and the number of bits needed for motion vector data. The motion-estimation technique that exhaustively

tests all possible motion representations is called a full-search motion-estimation. A faster method, which is sub-optimal with respect to rate-distortion, is to use a coarse search grid for a first approximation and to refine the grid in the surroundings of this approximation in further steps.

For the overlapped block-based motion-estimation, the pixel-wise prediction errors of a block and its overlapping neighboring blocks have to be weighted and summed according to the window function. In the process of successively finding/refining motion vectors, since some neighboring motion vectors may not be known yet, the corresponding prediction errors can be ignored. The major disadvantages of the overlapped block-based motion-estimation are increased computational complexity, and the fact that prediction errors and also the optimal motion vectors depend on neighboring blocks/motion vectors.

4.3 SGT: the Sub-Grouping Transformation Algorithm

The ME/C described in the previous section have two things in common: applying transform only on 2-D blocks in original or residual frames, and assuming uniform motion within a block or frame. The novelty of the proposed SGT approach is to reduce temporal redundancy through transform in temporal direction, and analyze motion at the pixel level instead of the frame/block level.

Figure 4.2 shows the structure of the proposed 3-D video compression system. The input video frames are first grouped into groups of frames so that the difference between each frame in a group of frames is below the pre-determined threshold T_G. After applying the spatial wavelet transformation on all frames in the group of frames, each frame is divided into spatial frequency sub-bands. The group of frames becomes

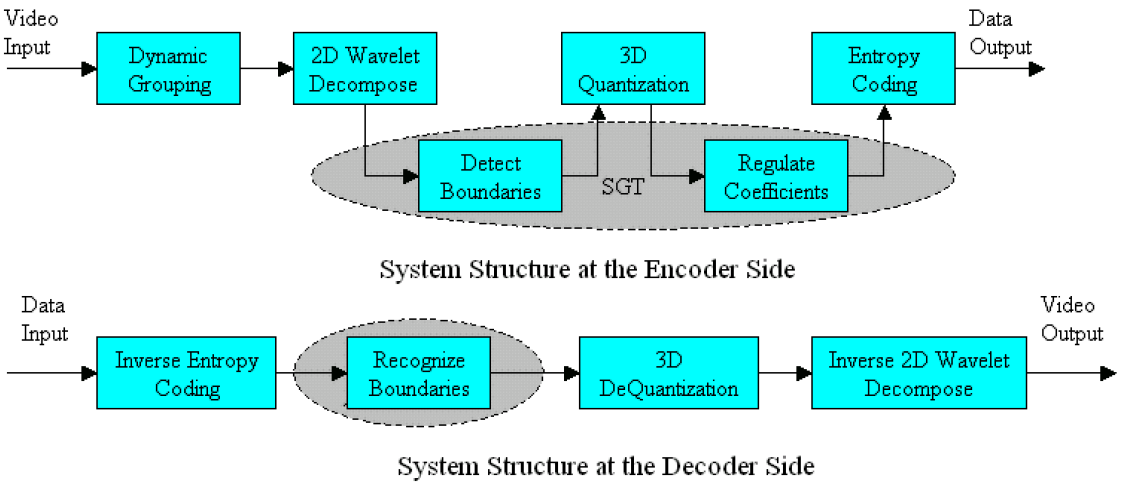


Figure 4.2: The structure of the proposed 3-D wavelet video compression system.

a 3-D matrix of spatial coefficients that are divided into different subbands in the spatial (x-y) domain (Figure 4.3). By far, it is the same as the conventional DWT-based video compression [58]. But, in the conventional 3-D DWT-based compression, the next step is to uniformly apply the temporal direction DWT on each 1-D array of coefficients along the temporal direction, while the proposed SGT algorithm treats each 1-D array differently. It evaluates these 1-D arrays of coefficients individually and divides each of them into smaller arrays (sub-groups) of variable sizes if necessary.

This approach is further illustrated in Figure 4.4. Picking an array at spatial location (x_1, y_1), the wavelet coefficients in this array could be very different in magnitude. If we draw this array in a plot, the plot may present a rapid fluctuated pattern globally, but have portions that are locally flat or linear increasing/decreasing. If we ignore the different local patterns of these arrays and indiscriminately apply the temporal DWT on all of them just like the conventional DWT-based compression, the resulting coefficients will include more and larger high frequency coefficients. These

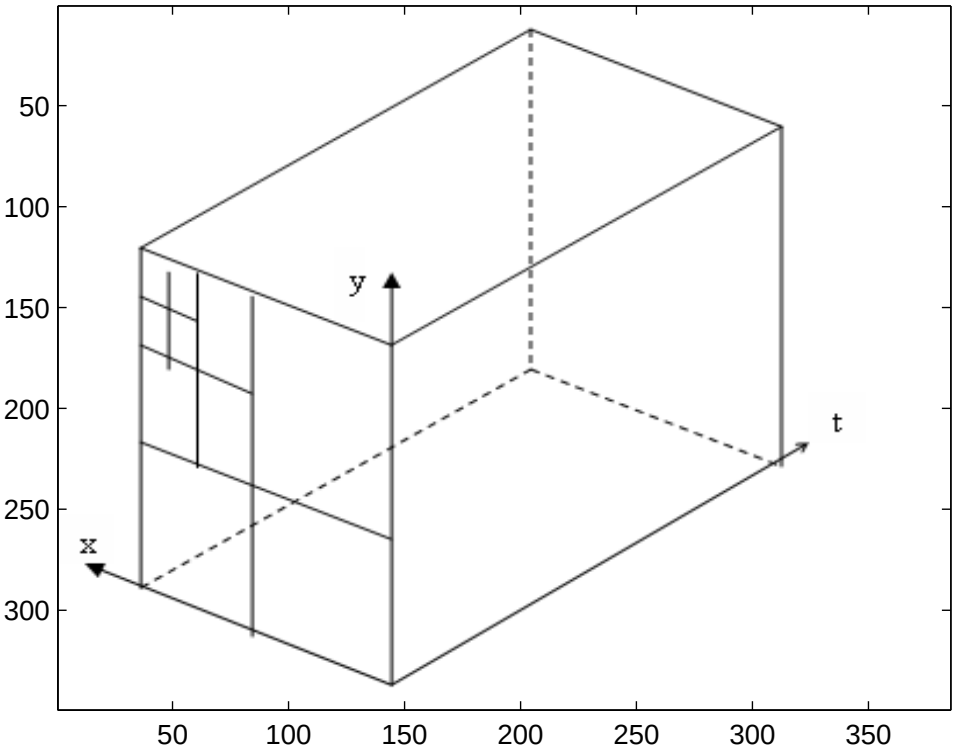


Figure 4.3: The 3-D matrix of spatial coefficients of a group of frames after spatial DWT.

large high frequency coefficients will still remain non-zero values if small quantization step/threshold is used, which will decrease the compression ratio; or become zeros if large quantization step/threshold is used, which will degrade the compression quality. To solve this problem, SGT algorithm divides an array at the borders of these local regular portions to get smaller arrays called sub-groups. The coefficients in each sub-group will then have a pattern that is thus flatter (less difference in magnitude) or more linear comparing with that of the entire original array. Instead of on the entire coefficient array, the 1-D temporal wavelet transform is applied within each sub-group. Because the change of the wavelet coefficients are relatively smoother within each sub-group, the temporal high frequency coefficients will have small magnitudes. Thus, they are more likely to be quantized to zeros with small quantization steps/threshold. Compared with applying temporal DWT on the entire original array, the SGT algorithm will have less non-zero coefficients after quantization with less quantization distortion. This means that it can achieve a higher compression ratio with the same quality, which is the goal that all motion-compensation algorithms are pursuing.

The two key steps of the SGT algorithm are: 1) finding the boundaries between sub-groups at the encoder and 2) recognizing the boundaries at the decoder. The algorithm should be accurate, efficient and robust. To be "accurate", the SGT algorithm must define the boundaries at the optimal locations at the encoder side and perfectly recognize them at the decoder; to be "efficient", SGT should not require too much computation and use less bits; to be "robust", SGT must not be interfered by lossy encoding operations, such as quantization.

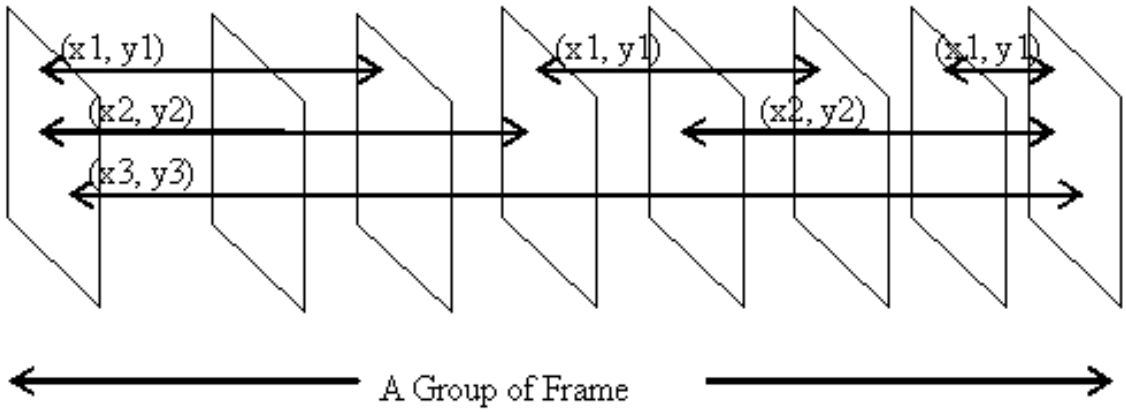


Figure 4.4: Wavelet coefficients are regrouped within a group of video frames, such as the coefficient arrays $(x1, y1)$, $(x2, y2)$, and $(x3, y3)$ shown here. The coefficient array $(x1, y1)$ is regrouped into three sub-groups, while $(x2, y2)$ is divided into two and $(x3, y3)$ remains in one.

4.3.1 Detect Boundaries

This step defines the boundaries at the optimal locations at the encoder side. The "optimal" here means that these boundaries should break the array into sub-groups that will generate the least non-zero coefficients after quantization, which is the requirement of high compression ratio. The search of the boundaries is based on singularity detection. Here we define "singularity" as the second-order Haar wavelet coefficients whose absolute values are beyond a threshold T_S. Undecimated Haar wavelet transform is applied twice on a 1-D array to get the second-order Haar wavelet coefficients. The positions where singularities happen are the potential boundaries for sub-groups. The reason to use second-order transform is to treat linear portions of the array as smooth.

Sub-grouping is not necessary for every coefficient array. For an array with little motion, keeping it as a single group is more efficient. For one with extreme motion, it is also better to leave it as a single group since dividing it will not generate low motion sub-groups. Also, to ensure the efficiency of wavelet transformation, a sub-group should not be too small. With the above considerations, sub-grouping is regulated by following rules.

A position p is defined as a sub-group boundary if and only if it satisfy these two criteria:

1. $|c_p| > T_S$, where $|c_p|$ is the absolute value of the second order Haar wavelet coefficient at position p, and T_S is the minimum threshold for being considered as a singularity [55]. The positions where singularities happens are chosen as candidates of the boundaries.

2. $L_p > T_L$, where L_p is the length of the sub-group to be created if p is a boundary and T_L is the minimum length of a sub-group.

The choice of T_S and T_L will affect the performance of the algorithm. While T_S might be dependent on the current coefficient array, T_L could be an independent parameter. Currently, we set T_S equal to the standard deviation of the original coefficient array and T_L equal to 8. Further investigation is going on to find better parameters.

4.3.2 Recognize Boundaries

To be efficient, the SGT should not store the locations of the sub-group boundaries in the output data stream as overhead. The decoder must automatically detect the boundary of each sub-group. This is successfully achieved by manipulating the coefficients following certain rules at the encoder side. The manipulation can only

be applied after the coefficients have been quantized. The reason for that is to avoid the interference of quantization/dequantization which may change the value of a manipulated coefficient.

At the encoder side, let B be a boundary coefficient, c a non-boundary coefficient, and s the standard deviation of the non-boundary coefficients. Because s is no more than 5 in most cases, we use 5 to replace s in computation. This not only makes it simple, but also make the results robust. The manipulating algorithm is illustrated by the pseudocode below.

1. Regulate non-boundary coefficients at the encoder side:

```
If |c|<= 2s
    c = c;
Else
    If c is even
        c = c;
    Else
        |c| = |c| - 1;
    End
End
```

   All non-boundary coefficients are forced to be even if it is bigger then $2s$. This will introduce a little bit of distortion on these large non-boundary coefficients. But the non-boundary coefficients belong to high frequency subbands, and their distribution density function, compared with Gaussian, has much higher peak at the mean value point which is around zero. So it is very safe to say that

the percentage of the non-boundary coefficients larger than $2s$ is less than 4%. Among them, only the odd ones need to be manipulated, which is 2%. The maximum possible distortion caused by this regulation on each odd large coefficient is less than 10%. Assuming that the odd large coefficients take 10% of the total absolute value sum, the maximum total distortion will be less then $10\% * 10\% = 1\%$. In fact, the above is based on very loose assumption, the total distortion is even smaller.

2. Regulate the boundary coefficients at the encoder side:

   (a) Small $|B|$

```
If |B| <= s
    |B| = 29;
Else if |B| <= 2s
    |B| = 19;
End
```

   (b) Large $|B|$

```
If |B| > 2s
    If B is even
        |B| = |B| - 1;
    Else
        B = B;
    End
    If |B| = 19, |B| = 17;
    //19 is reserved
```

```
        If |B| = 29, |B| = 27;
        //29 is reserved
    End
```

All boundary coefficients are forced to be odd. Because the boundary coefficients belong to low frequency subbands, they are typically much larger than $2s$. So the possibility that a) is triggered is so small that the distortion introduced by the regulation of it can be ignored. As for b), the distortion here is no more than 5% on each even B, total is no more than 2.5% for all boundary coefficients. Since boundary coefficients are holding the energy information, such a small distortion on them has little impact on the quality of the reconstructed video.

3. Recognize boundaries at the decoder side (c is an arbitrary coefficient):

```
If c is odd,
    If |c| = 29
        |c| = (s+1)/2 = 3;
    else if |c| = 19
        |c| = (2s+6)/2 = 8;
    End
    c is a boundary coefficient;
End
```

Manipulating coefficients introduces a little bit of distortion, but the quality of the reconstructed video is not really affected. Compared with the conventional motion-compensation methods which have to spend a lot of bits to record the motion trajectories, the proposed new method is more efficient and suitable for the very low bit rate video compression.

4.4 Sub-Grouping Transform with Intelligent Clustering

The border recognition method above is simple but has two problems:

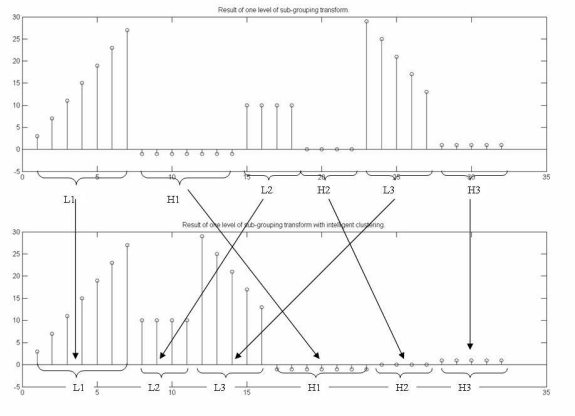
- The manipulating of boundary coefficients will cause some distortion.
- The non-zero low frequency coefficients of sub-groups will be located across the array, which may prevent the forming of long streams of zeros. This will reduce the efficiency of entropy coding.

4.4.1 Intelligent Clustering

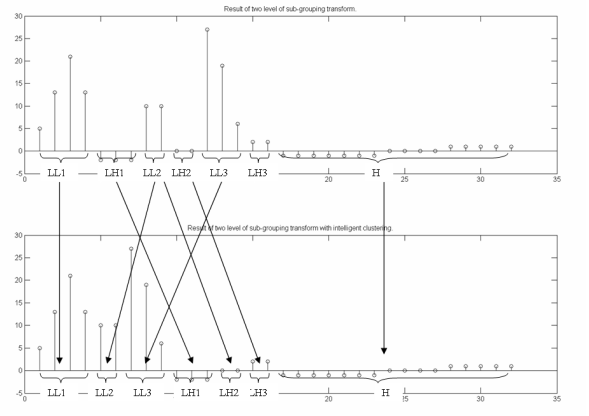
In this subsection, we propose another method that does not have these disadvantages. This proposed method is named Intelligent Clustering. The intelligent clustering algorithm is applied together with temporal transform after the sub-groups have been defined. After each transformation, the intelligent clustering algorithm moves the newly generated lower frequency subbands of all of the sub-groups together and put them in front of the newly generated higher frequency subbands. The benefit of this method is that lower frequency coefficients are clustered in the beginning of the coefficient array, which will increase the chance of creating long zero streams, thus to improve the efficiency of entropy coding.

Figure 4.5 gives an example of how the intelligent clustering algorithm works together with sub-grouping transform. In each figure, the upper plot is the coefficients after one level of sub-grouping transform and the lower plot is the result after intelligent clustering. The example array is divided into three sub-groups. In Figure 4.5(a), the input is the coefficients after the first level of sub-grouping transform. We can see that the lower frequency subbands and higher frequency subbands are located within their own sub-group. The intelligent clustering algorithm moves the lower frequency subbands to the left of the array and shifts the higher frequency subbands to the right. Using the term *frequency index* defined in the previous chapter, the array now has two parts. The left part includes subbands of frequency index 1 and the right part includes subbands of frequency index -1. Those subbands of the same frequency index still maintain their orders, i.e., the subband of sub-group i is in front of the subband of sub-group $i + 1$. In Figure 4.5(b), the input is the coefficients after the second level of sub-grouping transform. This transform only affects the subbands of frequency index 1. So we keep the portion of frequency index -1 unchanged and apply the intelligent clustering only on the other part. This process continues until all of the lowest frequency subbands only have one coefficient each. Figure 4.5(d) gives the final results. All of the subbands are sorted from left to right by their frequency indexes first, then by their origins in sub-groups. This method keeps the advantage of sub-group transform (less large high frequency coefficients), while avoiding its disadvantage (discontinuous zeros).

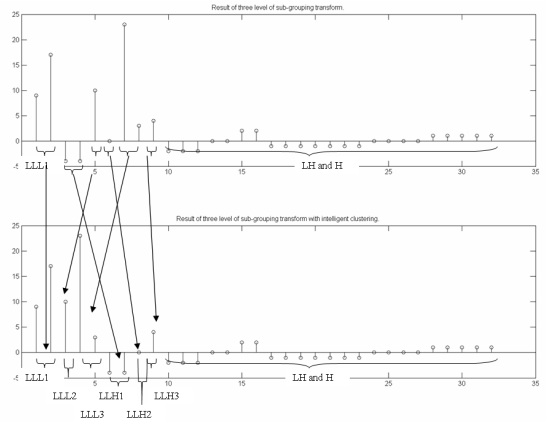
In Figure 4.6(a), from top to bottom, the first plot is the original array of coefficients, the second is the array of coefficients after temporal transform in the entire array, the third is the array of coefficients after sub-grouping transform, and the last



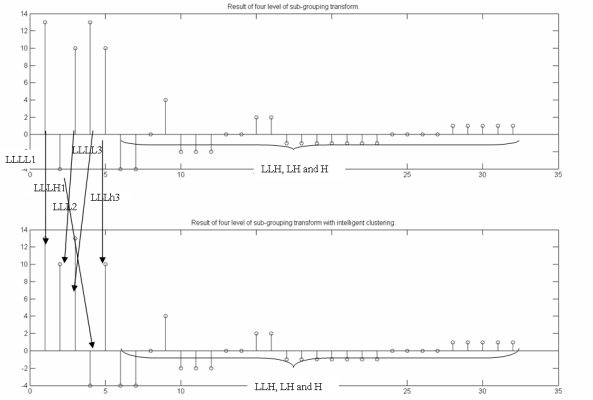
(a) One level of sub-grouping transform with or without intelligent clustering.



(b) Two levels of sub-grouping transform with or without intelligent clustering.



(c) Three levels of sub-grouping transform with or without intelligent clustering.



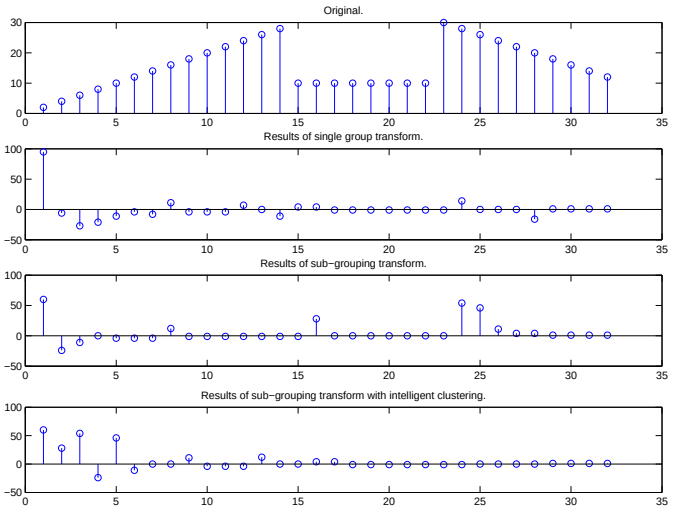
(d) Four levels of sub-grouping transform with or without intelligent clustering.

Figure 4.5: Sub-grouping transform with or without intelligent clustering.

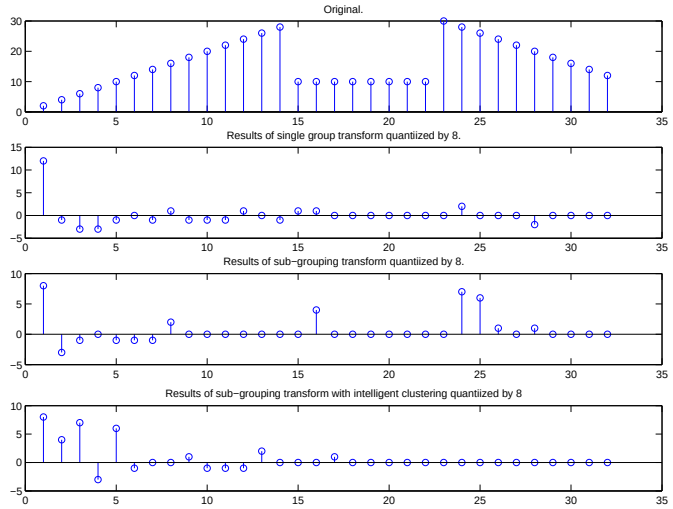
is the array of coefficients after sub-grouping transform with intelligent clustering. Each plot in Figure 4.6(b) corresponds to a plot in Figure 4.6(a), and the difference is that their coefficients are quantized. From the second plot, we can see that applying transform on the entire array of coefficients generated *sixteen* non-zero coefficients that are not continuous. To code these coefficients, we need *twenty one* run-length symbols. The third and the last plots both used sub-grouping transform and generated *twelve* non-zero coefficients. Without intelligent clustering, the non-zero coefficients are located at the left ends of each sub-group and all the zeros are pact in three groups. Thus, we need *seventeen* run-length symbols to represent them. With intelligent clustering, all the non-zero coefficients are located at the left side of the array and those zero coefficients have better continuousness. Thus we need only *fifteen* run-length symbols to represent them. In summary, sub-grouping transform has better entropy coding efficiency, and intelligent clustering can further improve it.

4.4.2 Recording the boundaries

To decode the data after applying the intelligent clustering algorithm, the boundaries of sub-groups have to be known by the decoder. The manipulation method methods previously is not suitable since the intelligent clustering method has changed the positions of the boundary coefficients. We designed a new method that is able to recover the sub-group information by recording the lengths of sub-groups of each array. This information is stored as overhead in the header of the compressed group of frames. The intelligent clustering algorithm improves the efficiency and quality of compression, so a little bit of overhead is affordable.



(a) Before quantization.



(b) After quantization.

Figure 4.6: Compare the different temporal transform methods. SGT with intelligent clustering has the best energy compact effect.

Assuming that the length of the group of frames is N, and the length of the ith sub-group is L_i where i is the index of the sub-group. Since the minimum length of a sub-group is T_L, we record the length of a sub-group as $l_i = L_i - T_L$, from which L_i can be easily recovered. Also because of the constrain of T_L, the maximum length of a sub-group is $L_i = N - T_L$, so the maximum of l_i is $N - 2 * T_L$. All numbers are converted to binary format before writing them as bits in the bit stream. So the length of the first sub-group can be recorded by at most $Nb_1 = \log_2 (N - 2T_L) + 1$ bits. After the length of the first sub-group is known, the length of the second sub-group can be recorded by at most $Nb_2 = \log_2 (N - 2T_L - L_1) + 1$ bits. The rest can be determined by the same method.

The details of recording the sub-grouping information is described below:

If $L_1 = N$ % This array is not divided into sub-groups
    Write '0' into the bit stream.
    Go to the next array.
Else
    While (TRUE)
        If i == 1
            Write '1' into the bit stream. % Start an array with sub-groups
        End
        $n = Nb_i - \log_2 l_i + 1$.
        If $n > 0$
            Write n '0's in the bit stream.
        End
        Write bits for l_i into the bit stream.
        If Sub-group i+2 is the last sub-group
            $n = Nb_{i+2} - (\log_2 l_{i+2} + 1)$.
            If $n > 0$
                Write n '0's in the bit stream.
            End
            Write bits for l_{i+2} into the bit stream.
            Go to the next array.
        Else If Sub-group i+1 is the last sub-group
            Write n '0's into the bit stream.

```
                Go to the next array.
            End
        End
    End
```

4.5 Experimental Results

To compare the performance of our proposed algorithms with those of others, we compressed the sample video clip with four algorithms: 3-D zerotree compression [57], the basic 3-D wavelet compression, sub-grouping transform, and sub-grouping transform with intelligent clustering. Arithmetic coding was used as entropy coding for the one using the zerotree algorithm, while Stack-run [44] and Arithmetic coding were used for the other three. The sub-grouping transform and intelligent clustering algorithms are only applied on the lower spatial frequency quarter of the group of frames since the rest higher spatial frequency portion does not have a lot of non-zero coefficients and applying sub-grouping transform on them will not have any benefit.

We compared the performances of the four video compression methods by their visual qualities at the same compression ratios. The one with the highest visual quality will certainly have the best performance. Because not all of them can precisely control the output bit rates, we make the compression ratios of our proposed algorithms higher than those of the opponents. It will be more convincing that the performances of the proposed algorithms are better if they have both better quality and higher compression ratios.

Since reducing ghost effect is an important goal of the proposed algorithm, the ghost effects of the compressed videos were evaluated separately. There is no objective method to measure the ghost effect, so we used subjective scores (human judgement)

Algorithm	Ghost Effect	Comp. Ratio	PSNR	WNMSE
3-D wavelet	visible	218	30.45	29.33
zerotree	visible	220	30.63	29.46
SGT	invisible	222	31.20	30.24
SGT+IC	invisible	220	31.24	30.50

Table 4.1: Compression results of the 10TV video clip.

given by human observers to indicate it. Two objective quality indexes, the Peak Signal-to-Noise Ratio (PSNR) and the Weighted sum of Normalized Mean Squared Errors (WNMSE) were used side by side to measure the overall visual quality of the reconstructed videos, where WNMSE can better match the visual quality evaluation of human visions.

Figure 4.7 uses an example frame to compare the performance of the SGT, SGT with IC (intelligent clustering), and zerotree algorithms. The ghost effects of the frame compressed by the SGT and SGT with IC are almost invisible, while that compressed by the 3-D zerotree compression has obvious ghost effect. Table 4.1 lists the complete experimental results. From this table, we can have a better view of the overall performance of the propose algorithms. With even higher compression ratios, the proposed algorithms has better visual quality and less ghost effect than the basic 3-D wavelet compression and the 3-D zerotree compression. The SGT with IC has slightly less compression ratio and better quality than SGT. It overall performance outperforms SGT mainly because it reorganizes the coefficients and thus improves the efficiency of stack-run coding, which compensates the overhead caused by storing the sub-group information. Also, using the intelligent clustering instead of manipulating the boundary coefficients avoids extra distortion.

(a) The original sample frame.

(b) The reconstructed sample frame compressed by the proposed SGT algorithm, compression ratio = 222.

(c) The reconstructed sample frame compressed by the proposed SGT algorithm with IC (intelligent clustering), compression ratio = 220.

(d) The reconstructed sample frame compressed by the 3-D zerotree algorithm, compression ratio = 220.

Figure 4.7: We can see that (b) and (c) has both higher compression ratios and obviously better visual quality than (d), while (c) has better quality than (b). Especially, the ghost effects of (b) and (c) are almost invisible, but (d) have obvious ghost effects at the top right side of the head.

4.6 Conclusions

This work exploits a new direction in reducing the temporal redundancy in 3-D DWT-based video coding. The results are encouraging. Without motion-estimation and motion vectors, the SGT algorithm achieves the goal of motion compensation by dividing wavelet coefficients in the temporal domain into smaller sub-groups and applies wavelet transform within each sub-group. Compared with other video coding algorithms with motion-compensation, SGT has better visual quality with the same bit-rate, especially for applications requiring very low bit rates.

Two methods are tried to recover the sub-grouping data. One recognizes boundary coefficients by manipulating their values, which is simple and does not use overhead bits, but it introduces extra distortion and can not be used with the intelligent clustering algorithm. The second method is more complex and codes the sub-grouping information as overhead in the header of a group of frames, but it makes stack-run coding more efficiency and does not cause extra distortion.

CHAPTER 5

3-D VIRTUAL SUB-OBJECT CODING

5.1 Introduction

Video coding algorithms can be divided into two categories: the block-based and object-based. The basic coding unit of block-based video coding systems is "block" that is not a natural representation of visual objects and causes the block artifacts for low bit-rate coding. The conventional video coding standards, such as H.261, H.263, MPEG-1 [59], MPEG-2 [60, 61], are all block-based. The MPEG-4 standard [62] represents the new object-based framework for efficient multimedia representation and enables content-based functionality by introducing the object-based coding. The object-based video coding codes semantic objects directly instead of rectangular blocks in video sequences. The main reason for switching to object-based coding is that images are naturally composed of visual objects. The conventional pixel-level description of images is only due to the lack of suitable tools to efficiently describe visual objects. Once objects have been identified and described, they can be treated individually for variant needs. In many applications, the object-based coding is obviously the most reasonable choice.

Many object-based coding techniques have been proposed in the literature. These techniques first segment video frames into a set of arbitrarily shaped moving objects and the background, and then code their shapes, motions, and textures. Some of the major advantages of the object-based coding include:

1. More accurate motion estimation of the moving objects.
2. Better utilizing the available bit-rate by only focusing on the moving objects.
3. Supporting content-based functionality such as video retrieval.
4. Encoding/Decoding selectively at different quality and resolution for each object.
5. More tolerable quality degradation with respect to human perception than the block-based approaches.

In terms of coding efficiency, the object-based coding presents some costs that do not appear in conventional block-based coding systems. First of all, since objects are separate entities, their shapes and locations must be described and sent to the decoder in advance as side information. Second, most coding techniques become less efficient when dealing with regions of arbitrary shapes, such as the reduced energy compaction of transformations. Finally, each object may need its own set of coding parameters, which adds on the cost of side information. On the positive side, an accurate segmentation actually carries with it the information on the graphical part of the image, i.e., the edges, and hence contributes to the quality of the reconstructed image. But an accurate segmentation is very expensive.

To deal with the temporal redundancy, object-based video coding can mimic the motion estimation/compensation mechanism of the conventional block-based video coding by replacing the reference blocks with the reference objects. The encoder extracts the reference objects, finds the motion vectors through motion estimation for the objects in the following frames, and reconstructs the coded frames to get the residual frames. Finally, it encodes the reference objects, their shapes and locations, motion vectors, and residual frames. Another approach of object-based video coding is based on 3-D wavelet coding while the group of frames being replaced by the group of video object planes (VOP), a 3-D object. A VOP is defined for object based coding to represent either a rectangular-plane frame or arbitrary-shaped object in a frame. The encoder extracts the VOPs, calculates the motion vectors by motion estimation, constructs the 3-D objects using the motion vectors, and finally codes the 3-D objects and their shapes, motion vectors, and background frames into bits. In this work, we will take the second approach and the focus is on the mechanisms that efficiently construct the 3-D objects. The proposed 3-D virtual sub-object coding is best suitable for videos with fixed background. This algorithm has a global motion trajectory for the virtual 3-D object. By dividing the virtual object into nine sub-objects, it in fact applies locally refined motion estimations. So its motion estimation takes into account of both the global and local optimization to improve the coding efficiency.

The chapter is organized as follows: Section 5.2 gives an overview of the object-based video coding technology; Section 5.3 will describe the proposed virtual sub-objects coding algorithm; Section 5.4 will compare the proposed algorithms with others using experimental results; Section 5.5 will summarize this chapter.

5.2 Overview of Object-based Video Coding

Object-based video coding includes video segmentation, motion estimation, and 3-D object coding, which are introduced below.

5.2.1 Video Segmentation

Video Segmentation can be defined as a process which typically partitions the video images into meaningful objects. Background can also be viewed as a special object. Approaches for segmenting video sequences into moving 3-D objects can be classified into four categories: spatial-temporal, motion, morphological, and model-matching techniques.

1. Spatial-temporal segmentation techniques attempt to identify the objects in a scene based on spatial and temporal information without explicitly computing the motion parameters [63] - [72]. The spatial information can be derived by measuring intensity or texture changes, while the temporal information can be generated by a change detection technique over multiple frames.

   Spatial segmentation is basically image segmentation, which partitions the frame into homogeneous regions with respect to their colors or intensities. This method can be typically divided into region-based and boundary-based. Region-based methods [73] rely on the spatial similarity in color, texture, and other pixel statistics to identify the "homogeneity" of these localized features. Boundary-based approaches use primarily a differentiation filter to detect the image gradient information and extract the edges. The discontinuous edges are then grouped to form the object contour. The main drawback of boundary-based

approaches is their lack of robustness during the contour closure extraction because of the difficulty in computing the region's closed boundaries. The spatial-based segmentation approach can provide more accurate object boundary than temporal-based method because of the high spatial correlation between the adjacent pixels within the object region. However, its relatively high computational complexity limits its application.

On the other hand, temporal segmentation, which is based on change detection followed by motion analysis, utilizes intensity changes produced by the motion of moving object to locate the position and boundary of objects in time and space. The most common motion information is the absolute difference between two consecutive frames. Unlike the spatial segmentation approaches, higher efficiency can be achieved because of the small number of operations for the segmented moving region instead of the whole image for every frame. However, lighting variation and noise might be incorrectly assigned to moving objects. It is usually very difficult to distinguish between changes due to true object motion and changes due to noise, shadow effects, and so on.

2. Motion segmentation techniques rely on motion parameters, explicitly computed from the spatial color and luminance information. Based on the motion parameters, each frame is segmented into a number of regions with coherent motion characteristics using various techniques such as the modified Hough transform [74], merging [75], Bayesian framework [69], [76], [77], and K-means [78]. Motion segmentation techniques, being tightly coupled with motion estimation, suffer from the two fundamental problems of occlusion and aperture, which affect the accuracy of the boundaries of segmented objects. Consequently, to overcome

these problems, several approaches were proposed to treat motion estimation and segmentation jointly, utilizing the Bayesian framework [79] - [81] or color, motion, and intensity change information [82], [83].

3. Morphological techniques [84] - [92], which involve morphological filters or watershed segmentation techniques, are computationally efficient. Morphological techniques typically start by a simplification step of the video frames using morphological filters. Then, a marker extraction step involves detecting the presence of homogeneous areas. Then, the undecided pixels are assigned a label in a decision step.

4. Model-matching segmentation techniques aim to locate the object in the video scene based on the best match between a model of the object and the frames. In general, a robust model-matching approach should address the issues of object occlusion, object deformation, and multiple moving objects in the presence of noise [93] - [97]. In [97], a partition of the feature space is first created. The training and learning phases are then used for the classifier. This method enables a combination of cues, such as texture, color, and depth. In order to achieve high classification accuracy, nonlinear decision functions are usually required when the sequences contain complicated content. The weakness of model-matching segmentation is that it need a training stage that is not automatical.

5.2.2 Motion Estimation

A 3-D object is the combination of a group of spatial objects called video object planes (VOP). Each of these VOP may cover different portion of the real moving

object, have different distances from the camera (different scales) and present different angles of the actual object. The efficiency of the 3-D object coding is largely dependent on how good the 3-D object can be constructed from its VOPs. This is in fact a motion estimation problem. The motions of arbitrarily shaped regions can be described by the parameters of an affine motion model. A well defined motion trajectory will organize the VOPs in a way that temporal redundancy can be efficiently exploited and thus the introduced distortion by compression is reduced.

Many object-based coding techniques for very low bit-rate video compression had been proposed in the literature. In [98], hierarchical block matching is used, which estimates displacements by different measurement window sizes, signal bandwidths, and maximum update displacements on several hierarchy levels. In [99], contrary to many object-based motion estimation algorithms, the proposed algorithm first estimates a dense motion field from the two successive original frames and then segments this motion field into homogeneous regions (objects) based on a two-dimensional affine motion model. In [100], the authors used a block-based motion estimation technique. For each 16×16 block inside a moving object, the algorithm searches a corresponding location in the previous frame to minimize the sum of absolute differences between the current and previous blocks. In [101], an object-oriented coding method using block-based motion vectors was proposed for detecting motion parameters that are robust to additive noise and abrupt motions. A model failure object compensation by fractal mapping of the residual image was also brought up. In [102], the 3-D object structure and the motion parameters are estimated simultaneously, where the motion is formulated as a nonlinear dynamic system whose state is represented by the motion parameters and the scaled depths of the object feature points. In [103],

the motion detection based on the frame difference is first applied to identify moving objects. Only for moving objects, motion vectors of the blocks inside the objects are estimated by a fast visual-pattern block-matching method.

The accurate motion representation is the key to the success of good motion compensation for coding purposes. However, traditionally, most of the object-based coding techniques suffer from the following problem: the segmentation and motion estimation techniques can prove to be computationally very expensive. Furthermore, the accurate representation of the shape of moving objects is the necessary condition to achieve the goal of good compression result. In contrast to the problems of object-based coders, the block-based low bit-rate video compression schemes do not suffer from the same drawbacks as the object-based coders. However, block-based video compression schemes also suffer from the problems of blocking artifacts, unnatural object motion at very low bit rates.

5.2.3 3-D Objects Coding

The difficulty of 3-D object coding is the *irregularity* of the shapes of 3-D objects. Because transformation is the key step of coding schemes, the difficulty of irregularity is in fact a problem of how to transform signals with arbitrary region of support (AROS). Many approaches have been proposed to address this problem. Just like the video processing technologies, the earliest work in this field was on images (2-D), and later some of the successful approaches were extended to videos (3-D).

In [105], the first method of transform with AROS was proposed to transform signal to Discrete Cosine Transform (DCT) domain. The attention on wavelet-based

coding is justified by the enormous success of this approach in image coding, leading to the new wavelet based standard JPEG-2000, and more recently video coding [106]. Many approaches have been proposed. Among them, the extension-over-the-boundaries method is the most popular one. Some of the extension methods are relatively simpler, such as filling up blocks not fully covered by the AROS [107] to a rectangular shape, or padding one-pixel at the end of the signal when the length of it is odd in the wavelet transform [108]. Although these methods are simple, they often result in overcomplete representation in the transform domain and reduce the coding efficiency. Critically sampled algorithms were thus preferred. Shape-adaptive DCT (SA-DCT) algorithms that involve pixel alignments before row and column transforms are proposed in [109], [110]. Similar algorithms for DWT using boundary extensions are studied in [111] - [113]. The Shape-Adaptive DWT (SA-DWT) proposed by S. Li and W. Li in [112] was the most recognized, laid the foundation for later researches, and was adapted as part of MPEG-4 standard.

The main coding tools of this work are the SA-DWT in [112], and a shape-adaptive version of 3-D Set Partitioning In Hierarchical Tree (SPIHT) [114, 49] proposed in [115]. We chose the method in [115] as our coding tools because SA-DWT is by now a de facto standard, and SPIHT guarantees a very good performance, and is widespread and well known in the compression community. Other coding algorithms based on shape adaptive wavelet have been proposed in recent years, such as the binary splitting with K-d trees (BISK) [116] algorithm that uses a simpler and more f1exible binary decomposition with K-d trees instead of the quad-tree structure used by most other algorithms.

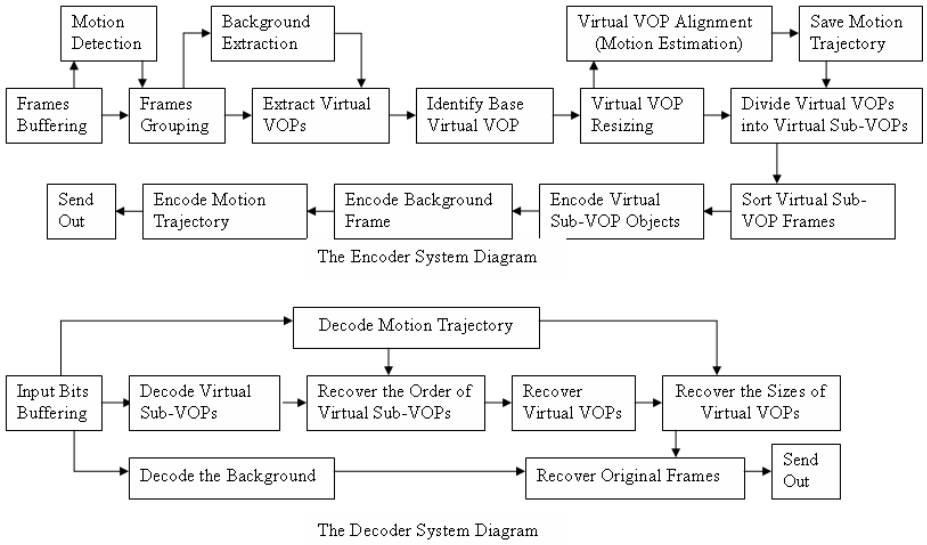


Figure 5.1: The encoding and decoding system diagrams of Virtual Sub-Object Coding.

5.3 3-D Virtual Sub-Object Coding (ViSC)

In this work, a new object-based video coding system is proposed, called 3-D virtual sub-object coding (ViSC). The purpose of ViSC is to find a fast and efficient compression approach for videos with fixed background. This system should have a motion estimation scheme that is optimized both globally and locally, generate as little side information as possible, and support the basic functionalities of object-based coding. The system diagram is displayed in Figure 5.1.

The incoming video frames are buffered before processing and the maximum buffer size is set to be *maxBuff*. The buffered video frames are processed as groups of frames

and the maximum group size is set to be *maxGroup*. In fact, *maxBuff* and *maxGroup* can be delay dependent, such as $maxBuff = 0.5 \times maxDelay \times frameRate$ and $maxGroup = 0.5 \times maxDelay \times frameRate$, respectively, where *maxDelay* is the upper limit of the playback delay and *frameRate* is the frame rate of the input video. Initially, the size of the first group is set to be *maxGroup*. Its background is first extracted from the frames. With this background, virtual VOPs are identified. Then the virtual-object based resizing and motion estimation are applied on these virtual VOPs to form a new group of frames with size *groupSize*. If $groupSize < maxGroup$, those $maxGroup - groupSize$ frames will be kept in the buffer to be used by next group. Virtual sub-objects will then be defined in this group and coded.

5.3.1 Background Extraction

Static background is the region that does not change between frames unless it is blocked by foreground objects. A spatial-temporal segmentation approach is used to extract the static background from the video sequence. Looking along the temporal direction, one pixel could belong to moving objects at some moments and background at the others. Because of the existence of noise, the same background pixel may have values with small variance on different frames. A temporal threshold *TemporalNoise* is defined to filter out the temporal noise interference and *SpatialNoise* to filter out the spatial noise interference.

First, we need to find the regions that are *pure* background. A pixel at position (x, y) belongs to the *pure* background only if it itself and all (if at the corner) or at least three of its four neighbors, $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$, $(x, y - 1)$, have variances smaller than *TemporalNoise* in the whole group of frames. It is possible

that a pixel is covered by moving objects all the time and satisfies this condition. Although this pixel is in fact a foreground pixel, there is no way to tell. So it is also classified as a *pure* background pixel, and this will not affect the algorithm. Having found the pure background regions, we evaluate the other locations, where one pixel is a background pixel only if

1. when compared with its nearest background neighbors, the difference is less than *SpatialNoise*;

2. when compared along the temporal direction with other pixels satisfying the first condition, the difference is less than *TemporalNoise*.

Finally, the "holes" left will be filled by repeating the surrounding background pixels into it (fake background area) to form the background frame. The entire background will be coded as a still image and sent out to the decoder for the first group. After that, only residue with respect to the previous background frame will be coded and sent out.

It is possible for the background to be improved gradually by new discovered *pure* background pixels until the whole *true* background is fully identified. A simple method is using a counter for each pixel. If a pixel stays with the same value (its changes are less than $max(TemporalNoise, SpatialNoise)$) and is identified as a background pixel continuously for more than *CountBG* times, this pixel will be set to its mean value and classified as a *true* background pixel, where *CountBG* is a predefined threshold value. A *true* background pixel will be stored and not be updated in the future unless a newer *true* value is available. The whole true background frame will be stored at both the encoder and decoder, so only the update data needs to be

sent to the decoder. But this requires additional memory and computation on the encoder side.

5.3.2 Virtual Sub-Object Construction

VOP Extraction

After the background is available, the boundaries of VOPs can be easily identified by subtracting the background from each frame followed by a change detection method that uses an inner-bound search with a change sensitive threshold to detect the boundaries. Each enclosed region with a size larger than *minVOPSize* represents a VOP on this frame. In the case that there are multiple 3-D objects, it is necessary to track their VOPs across frames, which is another topic that has been widely addressed. This work will only deal with the single 3-D object situation. VOPs are extracted frame by frame as long as the moving object does not leave the scene and the maximum buffer size is not reached. A rectangular virtual VOP that is just big enough to cover it is created for each VOP. Within this virtual VOP, pixels out of the VOP boundary are filled with zeros. These virtual VOPs along with their location information are stored in order to be used at the next step, motion estimation. Figure 5.2 gives an example of VOP extraction. The original frames are listed at the left side and the extracted VOPs are displayed at the right side.

Find the Base Virtual VOP

The distance and direction, with respect to the camera, of the moving object may be different for every video frame when they are captured by the camera. As a result, virtual VOPs represent the moving object with different scales and angles, and some of them may only cover one portion of the object. The virtual VOP that covers the

(a) Original Frame085

(b) Extracted VOP from Frame085

(c) Original Frame090

(d) Extracted VOP from Frame090

(e) Original Frame095

(f) Extracted VOP from Frame095

Figure 5.2: The original frames are listed at the left side and the extracted moving objects are shown at the right side.

smallest portion is called the *base* virtual VOP. The other virtual VOPs may not be at the same scale as the base virtual VOP because they may be at different distances from the camera. They may also cover larger portions of the moving object. The base virtual VOP is a subset of any other virtual VOP if they are at the same scale.

Normally, the closer the object moves to the camera, the larger its image will be in the video and the smaller the covered portion of the object if the object is not completely in the scene. If the object is in the scene, both its height and width will increase when it moves closer to the camera. Considering that the object may change its direction of motion, it is possible that it becomes "thinner" and higher. So the height is a better indicator of distance if the object is completely in the scene. The one with the maximum height is used as the base virtual VOP. If the heights are the same for more than one virtual VOPs then the narrower one will be used as the base virtual VOP. We will see later that all of the virtual VOPs in the same group will be resized to the same resolution as the base virtual VOP. If a resized virtual VOP is smaller than the base virtual VOP, it will become the new base virtual VOP to secure that the base virtual VOP is always a subset of all of the other virtual VOPs. In the situation where only a portion of the object is in the scene, the closer image may be shorter or higher, but wider for sure. Thus we use the widest virtual VOP as the base virtual VOP. If two virtual VOPs are the same in width, the shorter one will be chosen as the base virtual VOP. If one VOP has crossed the horizontal boundary of the scene at both sides, it will be considered "wider" than any other virtual VOP that is still within the horizontal boundary. There are exceptions that need to be taken care of:

1. Objects crossing the scene boundary. The VOPs that are moving in or out across the scene boundary will be treated separately since their dimensional changes do not obey the rules above. But this is not a big deal since it is easy to tell whether an object is moving in or out.

2. Objects changing their shapes. It is possible for an object to change its shape within the scene, such as a person who is waving his arms. This could be dealt by reordering the frames when it is "periodical" or beginning a new group if the change is dramatic. Otherwise, a singularity test will detect such changes and treats them as singular VOP. A virtual VOP with singular VOP can not be used as the base virtual VOP.

3. Objects changing their angles. There will be angle changes between virtual VOPs. The change of angle is usually in a smooth pace and does not affect much. Sudden change of angle will force the encoder to start a new group if it causes a big enough difference between the current and the previous virtual VOPs.

Globally and Locally Optimized Motion Estimation

With the base virtual VOP in hand, the next and the most important step is to find the optimal motion trajectory to link the VOPs together. The idea here is basically the same as the block-based motion estimation, i.e., utilizing the temporal redundance as much as possible so that the highest coding efficiency could be achieved. The main difference is that the motion estimation here is object-based (virtual VOP based to be precise) instead of block-based. It includes three steps:

(a) The virtual VOP of Frame085.

(b) The resized virtual VOP of Frame085.

(c) The virtual VOP of Frame090.

(d) The resized virtual VOP of Frame090.

(e) The virtual VOP of Frame095, the base virtual VOP.

(f) The virtual VOP of Frame095, the base virtual VOP.

Figure 5.3: Resize the virtual VOPs to match the resolution of the base virtual VOP, Figure 5.3(e). The original virtual VOPs with different resolutions are at the left side and the resized at the right side. Another copy of the base virtual VOP is list as Figure 5.3(f) for the purpose of easy comparison.

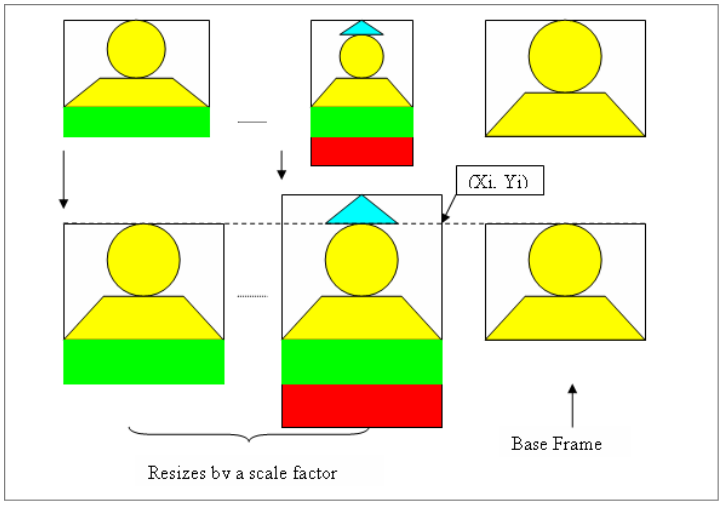


Figure 5.4: Resize and align the virtual VOPs.

1. Spatial resizing. This step enlarges the ith virtual VOP by a scale factor SF_i so that it has the same spatial resolution as the base virtual VOP. If a resized virtual VOP is smaller than the base virtual VOP, it will become the new base virtual VOP to secure that the base virtual VOP is always a subset of all the other virtual VOPs. The moving object may move towards or away from the camera even within a group of frames, which cause the fluctuation of the scale factors of virtual VOPs.

2. Motion estimation. This step matches the base virtual VOP with the other virtual VOPs to find the global motion trajectory. According to the definition of the base virtual VOP, the VOP of the base virtual VOP is a subset of those of all the other virtual VOPs. After the ith virtual VOP has been resized by

SF_i, a slide window of the same size of the base virtual VOP is applied on it from the top to bottom and left to right to cut a smaller virtual VOP VOP_{X_i,Y_i} from it, where (X_i, Y_i) is the coordinate of the top-left pixel of VOP_{X_i,Y_i} in the resized ith virtual VOP. Each VOP_{X_i,Y_i} will be compared with the base virtual VOP and find the one with the smallest mean squared error with respect to the base virtual VOP. The position (X_i, Y_i) of the best matched VOP_{X_i,Y_i} is then saved in the motion trajectory. Figure 5.4 shows a simple example of how the virtual VOPs are scaled and aligned. The resulting motion trajectory is a list of structure that records the following information of the virtual VOP in frame i, its position (x_i, y_i), its width and height w_i and h_i, its scale factor SF_i, and the match point (X_i, Y_i) of the base virtual VOP in this resized virtual VOP. The shape information is not necessary since we use virtual VOPs that are all rectangular.

3. Sub-Object generation. This step divides every virtual VOP into up to 9 virtual sub-VOPs. All of these sub-VOPs in the GOF are grouped into 9 virtual 3-D sub-objects. Each virtual VOP covers larger area than the base virtual VOP. Some may only extend to one side in the scene, and the other may extend to more than one directions. Since they are all aligned by the global motion trajectory with the base virtual VOP, we divide each virtual VOP into one to nine virtual sub-VOPs, depending on how they are different from the base virtual VOP. Each virtual sub-VOP is labeled according to its relative location to the base virtual VOP (Figure 5.5). The virtual sub-VOPs with the same label are grouped together, aligned to their "inner" borders or corners with respect to the base virtual VOP, and sorted by their sizes and overlap ratios

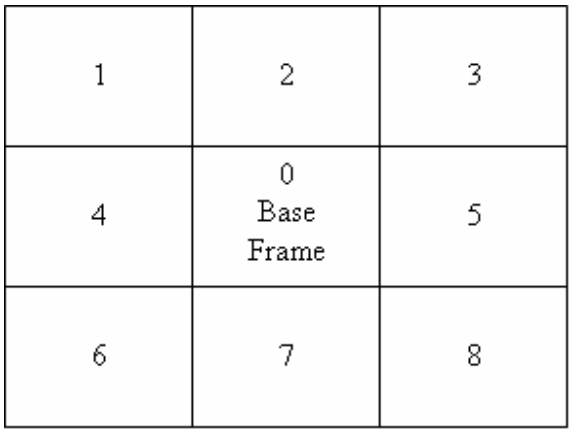


Figure 5.5: Virtual Sub-VOPs.

from largest to the smallest to form the virtual 3-D sub-objects. The sizes and orders of virtual sub-VOPs need not to be sent to the decoder because the decoder can figure out these parameters from the motion trajectory. Except for the one containing the base virtual VOP, the other virtual sub-objects may very likely contain less number of virtual VOPs than the number of the frames in the group, since not every virtual VOP extends to all the directions from the base virtual VOP. Compared with the virtual object method in [117], this method has better motion estimation and encodes less regions. So it is more efficient.

5.3.3 Virtual Sub-Object Coding

Strict 3-D SA-DWT based coding only transforms and codes the 3-D object (a group of VOPs). But the irregular shape of the VOPs reduces the coding efficiency and the shape information of VOPs adds additional side information to be coded. So the shape-adaptive approach is not always a good choice. The another option is to treat the group of virtual VOPs as a 3-D object and code this virtual object instead.

In this work, we apply the second idea to the nine virtual sub-objects we have generated, individually. Their shape information can actually be calculated from the parameter set $(x_i, y_i, w_i, h_i, SF_i, X_i, Y_i)$ that is recorded in the motion trajectory. This method will encode some unwanted areas, but it only has to code rectangular virtual VOPs so that the shape coding is avoided. The virtual sub-object 0 is in fact a well matched video sequence with very high temporal redundancy. So shape adaptive coding has little benefit on it. Instead, we used the conventional 3-D SPIHT algorithm on it since 3-D SPIHT fits well on its pattern. The other eight virtual sub-objects likely have frames with different sizes, but their spatial and temporal dimensions are known to both the encoder and decoder (derived from the motion trajectory). Among them, the virtual VOPs in sub-objects 2, 4, 5, 7 have either the uniform width or height, while those in the sub-objects 1, 3, 6, 8 have arbitrary width and height. The shape adaptive 3-D SPIHT coding is applied to these eight sub-objects but their rectangular shapes will not damage the efficiency of transform and there is no need for coding the shape information.

Video sequence	Bit rate (kbps)	Y/C	VOW	MPEG-4	ViSC
Akiyo	98	Y	33.59	33.71	33.42
		C	39.49	41.13	40.36
News	120	Y	32.50	32.95	31.37
		C	40.04	42.06	38.89
Coast	98	Y	28.98	28.74	28.04
		C	40.71	44.54	40.30
Cont	120	Y	29.42	29.26	28.58
		C	35.41	37.40	35.23

Table 5.1: Comparison of Virtual Sub-Object coding with VOW, MPEG-4 in PSNR with the same bit rates.

5.4 Experimental Results

Experiments were carried out to compare the performance of the proposed virtual sub-object coding (ViSC) algorithm with the shape-adaptive video object wavelet (VOW) coder [118] and MPEG-4. To be able to compare with VOW, we used the same video sequences the authors used in [118], where the objects are the woman in Akiyo (Akiyo), boat in Coastguard (Coast), anchor persons in News (News), and ship in Container (Cont). These sequences are in CIF resolution (352×288) and at 10 frames per second. The data of VOW and MPEG-4 were cited directly from Table II in [118].

Table 5.1 lists the results of comparison, where 'Y' and 'C' represent Luminance and Chrominance, respectively. The quality indexes, such as 33.59, are measured in PSNR (Peak Signal to Noise Ratio). The ViSC only has comparable performance with the other two coding algorithms for the Akiyo sequence. This is because the ViSC is designed to work with fixed background and currently only support single

moving object. Among the four sequences, only Akiyo fits the criteria. Note that, ViSC does not apply pre- or post-processing, while MPEG-4 must have applied, not sure about VOW though. Also, the lengths of the test sequences are not long enough for ViSC which is more suitable to long sequences than the other two.

5.5 Conclusions

In this work, we proposed an object-based 3-D wavelet coding algorithm called 3-D virtual sub-object coding that is suitable for videos with fixed background. In stead of allowing the arbitrary shaped objects, this algorithm uses a virtual object concept that constrains the shapes of objects to rectangular. This algorithm only uses one globally optimized motion trajectory for a virtual object. By dividing the virtual object into as many as nine sub-objects, it equivalently applies locally refined motion estimation that provides local optimization. Without using a complicated motion estimation and arbitrary shape adaptive transform, this algorithm achieved comparable performance compared with the state of the art 3-D wavelet based video compression algorithm VOW, and the leading standard MPEG-4.

The shapes of the virtual objects in this work are simple rectangular. This is the extreme option when considering the trade-off between the shape coding overhead and object coding bits. It is reasonable to assume that a simple and efficient description of the shape information will improve the performance. Also, a more sophisticated motion estimation that can track and describe the rotation of the object will also help.

CHAPTER 6

CONCLUSIONS

This dissertation covers the research topics in image quality assessment, image compression and video compression.

The concepts of human vision system (HVS) plays a very important role in the image and video technologies. A review of HVS is given right after the introduction. Since the HVS is so complex and much of it is still unknown to us, this review only covered those concepts that are relevant to our research or this dissertation in some way. The characteristics, especially the limitations, of HVS directed the research in lossy compression and quality assessment. This review mainly focused on them.

The wavelet transform is chosen as the tool through this research. This is because its spatial-temporal multi-resolution property is by far a good match for the main HVS characteristics. It can decompose images into compact frequency subbands with spatial correlation. This makes it a perfect tool for resolution and bit-rate scalable compression. Since wavelet has been widely studied and well known, this dissertation did not devote a chapter to it.

A image quality metric WNMSE was proposed which is based on the concepts of HVS and utilizes the properties of the wavelet transform. WNMSE uses the weighted sum of the normalized mean square errors of wavelet coefficients to assess the quality

of an image. According to the concepts of HVS, the weight for each subband is chosen to reflect its perceptual impact on the image, which measures the distortions in the global structure and local details of an image in a more balanced way automatically. Because WNMSE is defined in the wavelet domain, it can be calculated in the middle of compression without reconstructing the image. Furthermore, it facilitates the link between the quantization steps and the quality metric.

A quality constrained compression algorithm QCSQ was proposed. Based on the analysis of the relationship among the subband features, steps, and WNMSE values, we also invented a quality constrained compression algorithm QCSQ which can identify the quantization step step-size for every subband of an image. With these steps, the image can be compressed to a desired visual quality measured by WNMSE.

A new temporal filter scheme SGT was developed. Without motion-estimation and motion vectors, the SGT algorithm achieves the goal of motion compensation by diving wavelet coefficients in the temporal domain into smaller sub-groups and applies wavelet transform within each sub-group. Two methods are tried to recover the sub-grouping data. One recognizes boundary coefficients by manipulating their values, which is simple and does not use overhead bits, but it introduces extra distortion and can not be used with the intelligent clustering algorithm. The second method is more complex and codes the sub-grouping information as overhead in the header of a group of frames, but it makes stack-run coding more efficiency and does not cause extra distortion.

At the last, we proposed an object-based 3-D wavelet coding algorithm called 3-D virtual sub-object coding that is suitable for videos with fixed background. In stead

of allowing the arbitrary shaped objects, this algorithm uses a virtual object concept that constrains shapes of the objects to rectangular. This algorithm only uses one globally optimized motion trajectory for a virtual object. By dividing the virtual object into nine sub-objects, it, in fact, applies locally refined motion estimations that provides local optimization. Without using a complicated motion estimation and arbitrary shape adaptive transform, this algorithm achieves comparable performance compared with the state of the art 3-D wavelet based video compression algorithm VOW, and the leading standard MPEG-4.

APPENDIX A

QCSQ ALGORITHM IMPLEMENTATION DETAILS

A.1 Find the Initial Set of Quantization Steps

$s_{b_l} = C_l \cdot V_{b_l}$

1. Find the initial set of quantization steps for subband b_l in the level-l, $x \in \{h, v, d\}$ and $l \in \{1, 2, ..., L\}$.

$C_l = 4^{(L-l)}$;
if $vm_{a_1} > (0.7 + 0.1 \times l)$
    $V_{b_l} = \text{ceil}(\sigma_{b_l}) \times 2^{-f_{b_l}/2}$;
else if $vm_{a_1} < 0.2$
    $V_{b_l} = \text{floor}(\sigma_{b_l}) \times 2^{-f_{b_l}/2}$;
else if $vm_{a_1} > 0.6$
    if $(m_{a_1} > 96)$
        $V_{b_l} = \text{ceil}(\sigma_{b_l}) \times 2^{-f_{b_l}/2}$;
    else
        $V_{b_l} = \text{round}(\sigma_{b_l}) \times 2^{-f_{b_l}/2}$;
    end
else if $(m_{a_1} > 96)$
    $V_{b_l} = \text{round}(\sigma_{b_l}) \times 2^{-f_{b_l}/2}$;
else
    $V_{b_l} = \text{floor}(\sigma_{b_l}) \times 2^{-f_{b_l}/2}$;
end.
$s_{b_l} = C_l \times V_{b_l}$;
if $s_{b_l} < 1$
    $s_{b_l} = 1$;

else if $s_{b_l} > 256$
    $s_{b_l} = 256$;
end.

2. Find the initial set of quantization steps for subband a_L.

$C_L = 4^{(L-L)} = 1$;
if $m_{a_1} < (160 - 32 \times L)$
    $V_{a_L} = \text{floor}(\sigma_{a_L}) \times 2^{(NH_{a_L} - NL_{a_L})/2}$;
else if $vm_{a_1} > (0.7 + 0.1 \times l)$
    $V_{a_L} = \text{ceil}(\sigma_{a_L}) \times 2^{(NH_{a_L} - NL_{a_L})/2}$;
else if $vm_{a_1} < 0.2$
    $V_{a_L} = \text{floor}(\sigma_{a_L}) \times 2^{(NH_{a_L} - NL_{a_L})/2}$;
else if $vm_{a_1} > 0.6$
    if $(m_{a_1} > 96)$
        $V_{a_L} = \text{ceil}(\sigma_{a_L}) \times 2^{(NH_{a_L} - NL_{a_L})/2}$;
    else
        $V_{a_L} = \text{round}(\sigma_{a_L}) \times 2^{(NH_{a_L} - NL_{a_L})/2}$;
    end
else if $(m_{a_1} > 96)$
    $V_{a_L} = \text{round}(\sigma_{a_L}) \times 2^{(NH_{a_L} - NL_{a_L})/2}$;
else
    $V_{a_L} = \text{floor}(\sigma_{a_L}) \times 2^{(NH_{a_L} - NL_{a_L})/2}$;
end.
$s_{a_L} = C_L \times V_{a_L}$;
if $s_{a_L} < 1$
    $s_{a_L} = 1$;
else if $s_{a_L} > 256$
    $s_{a_L} = 256$;
end.

The specific values used in this algorithm are first chosen according to the rules described above, and finally determined after adjustments using experiments. Our experimental results show that these values are independent of wavelets used and suitable for all kinds of natural images.

A.2 Tune the Initial Quantization Steps

The initial set of quantization steps has quantized the image to a W-NMSE equal to Q_0, and the objective W-NMSE is Q. The difference is $\Delta Q = Q$ - Q_0. Assuming that subband α has the highest optimality opt_α and subband β has the second highest optimality opt_β, and their quantization steps, the averages of the quality gains, and the standard deviations of the quality gains are (s_α, m_α, std_α) and (s_β, m_β, std_β), respectively. The error threshold is δ, that is, we call it a successful tuning if the difference between the achieved and the target quality indexes is less then δ.

The process includes three iterative steps:

1. Adjust quantization steps to improve image quality.

   if $\Delta Q > m_\alpha$
   reduce s_α by half;
   $\Delta Q = \Delta Q$ - m_α;
   else if m_β / $opt_\beta > m_\alpha$ / opt_α
   reduce s_α by half;
   $\Delta Q = \Delta Q$ - m_α;
   else
   reduce s_β by half;
   $\Delta Q = \Delta Q$ - m_β;
   end.

2. Check whether the target quality is achieved. If not, go back to step 1); if yes, go to step 3).

   if $\Delta Q < 0$
   done;
   else
   $\alpha \leftarrow$ the subband (α or β) whose quantization step was not modified;
   $\beta \leftarrow$ the subband whose optimality level is next to β;
   repeat 1;
   end.

3. Calculate the current predicted quality metric Q'. If it is too big compared with Q, tune it down; if it is too small, go back to step 1).

while $Q' - Q > \delta$
recover the quantization step of the subband x that was the last being modified;
$Q' = Q' - m_x$, where m_x is the average quality gain of the subband x;
end.
if $Q - Q' < \delta$
$\alpha \leftarrow$ the most optimal subband among those whose quantization step was never modified;
$\beta \leftarrow$ the second most optimal subband among those whose quantization step was never modified;
repeat 1;
end.

BIBLIOGRAPHY

[1] R. A. Young, "Oh say, can you see? The physiology of vision," in *Proceedings of SPIE Human Vision, Visual Processing and Digital Display II*, vol. 1453, pp. 92-123, San Jose, CA, June 1991.

[2] A. A. Michelson, *Studies in Optics*, University of Chicago Press, 1927.

[3] E. Peli, "Contrast in complex images," *Journal of the Optical Society of America*, vol. 7, pp. 2032-2040, 1990.

[4] J. Lubin, "A visual discrimination model for imaging system design and evaluation," *Vision Models for Target Detection and Recognition*, pp. 245-283, World Scientific Publishing, 1995.

[5] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America*, vol. 2, pp. 1160-1169, 1985.

[6] C. F. Stromeyer and S. Klein, "Evidence against narrow-band spatial frequency channels in human vision: The detectability off requency modulated gratings," *Vision Research*, vol. 15, pp. 899-910, 1975.

[7] P. Vandergheynst, Ö. N. Gerek, "Nonlinear pyramidal image decomposition based on local contrast parameters," in *Proceedings of the Nonlinear Signal and Image Processing Workshop*, vol. 2, pp. 770-773, Antalya, Turkey, June 1999.

[8] O. Braddick, F. W. Campbell, and J. Atkinson, "Channels in vision: Basic aspects," *Handbook of Sensory Physiology*, vol. 8, Springer-Verlag, 1978.

[9] J. G. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vision Research*, vol. 20, pp. 847-856, 1980.

[10] A. B. Watson, "Temporal sensitivity," *Handbook of Perception and Human Performance*, vol. 1, John Wiley & Sons, 1986.

[11] R. F. Hess and R. J. Snowden, "Temporal properties of human visual filters: Number, shapes and spatial covariation," *Vision Research*, vol. 32, pp. 47-59, 1992.

[12] R. E. Fredericksen and R. F. Hess, "Estimating multiple temporal mechanisms in human vision," *Vision Research*, vol. 38, pp. 1023-1040, 1998.

[13] T. Betchaku, N. Sato and H. Murakami, "Subjective evaluation methods of facsimile image quality," in *Proceedings of IEEE International Conference on Communications*, vol. 2, pp. 966-970, May 1993.

[14] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *IEEE Transactions on Image Processing*, vol. 15, pp. 3441-3452, November 2006.

[15] B. Girod, "What's wrong with mean-squared error," in *Digital Images and Human Vision,* pp. 207-220, the MIT press, 1993.

[16] N. Damera-Venkata, T. D. Kite, W. S. Geisler, B. L. Evans, and A. C. Bovik, "Image quality assessment based on a degradation model," *IEEE Transactions on Image Processing*, vol. 9, pp. 636-650, April 2000.

[17] A. M. Eskicioglu and P. S. Fisher, "Image quality measures and their performance," *IEEE Transactions on Communication*, vol. 43, pp. 2959-2965, December 1995.

[18] Private Communication, N. Obuchowski to K. Powell, "Results of preference studies of compressed diagnostic breast images," Technical Report, *Biomedical Engineering, Cleveland Clinic Foundation*, February 3, 2000.

[19] Z. Wang, A. C. Bovik, H.R. Sheikh and E.P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, pp. 600-612, April 2004.

[20] A. Shnayderman, A. Gusev, and A. M. Eskicioglu, "An svd-based grayscale image quality measure for local and global assessment," *IEEE Transactions on Image Processing*, vol. 15, pp. 422-429, February 2006.

[21] A. B. Watson, J. Hu, and J. F. McGowan III, "DVQ: A digital video quality metric based on human vision," *Journal of Electric Imaging*, vol. 10, pp. 20-29, January 2001.

[22] M. Sendashonga and F. Labeau, "Low complexity image quality assessment using frequency domain transforms," in *Proceeding of IEEE International Conference on Image Processing*, pp. 385-388, October 2006.

[23] R. L. De Valois and K. K. De Valois, *Spatial vision*, Oxford University Press, 1988.

[24] K. Bao and X. Xia, "Image compression using a new discrete multiwavelet transform and a new embedded vector quantization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 833-842, September 2000.

[25] N. B. Karayiannis, P. Pai, and N. Zervos, "Image compression based on fuzzy algorithms for learning vector quantization and wavelet image decomposition," *IEEE Transactions on Image Processing*, vol. 7, pp. 1223-1230, August 1998.

[26] S. Kasaei, M. Deriche, and B. Boashash, "A novel fingerprint image compression technique using wavelets packets and pyramid lattice vector quantization," *IEEE Transactions on Image Processing*, vol. 11, pp. 1365-1378, December 2002.

[27] Y. Huh, J. J. Hwang, and K. R. Rao, "Block wavelet transform coding of images using classified vector quantization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, pp. 63-67, February 1995.

[28] P. C. Cosman, R. M. Gray, and M. Vetterli, "Vector quantization of image subbands: A survey," *IEEE Transactions on Image Processing*, vol. 5, pp. 202-225, February 1996.

[29] X. Wang, L. Chang, M. K. Mandal and S. Pancharathen, "Wavelet-based image coding using nonlinear interpolative vector quantization," *IEEE Transactions on Image Processing*, vol. 5, pp. 518-522, March 1996.

[30] E. A. B. da Silva, D. G. Sampson, and M. Ghanbari, "Super high definition image coding using wavelet vector quantization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 399-406, August 1996.

[31] A. B. Watson, G. Y. Yang, J. A. Solomon, and J. Villasenor, "Visibility of wavelet quantization noise," *IEEE Transactions on Image Processing*, vol. 6, pp. 1164-1175, August 1997.

[32] Z. Liu, L. J. Karam, and A. B. Watson, "JPEG2000 encoding with perceptual distortion control," *IEEE Transactions on Image Processing*, vol. 15, pp. 1763-1778, July 2006.

[33] M. J. Nadenau, J. Reichel, and M. Kunt, "Wavelet-based color image compression: Exploiting the contrast sensitivity function," *IEEE Transactions on Image Processing*, vol. 12, pp. 58-70, January 2003.

[34] A. Croisier, D. Esteban and C. Galland, "Perfect channel splitting by use of interpolation/decimation/tree decomposition tichniques," in *Proceeding of International Conference on Information Systems*, Patras, Greece, pp. 443-446, August 1976.

[35] J. W. Woods and S. D. O'Neil, "Subband coding of images," *IEEE Transactions on Communication*, vol. COM-31, pp. 532-540, April 1983.

[36] H. Jozawa, H. Watanabe, and S. Singhal, "Interframe video coding using overlapped motion compensation and perfect reconstruction filter banks," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 649-652, March 1992.

[37] H. Guo and C. Burrus, "Wavelet Transform based Fast Approximate Fourier Transform," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1973-1976, April 1997.

[38] Z. Xiong, K. Ramchandran, M. T. Orchard, and Y. Zhang, "A comparative study of DCT-based and wavelet-based image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 692-695, August 1999.

[39] R. W. Buccigrossi and E. P. Simoncelli, "Image compression via joint statistical characterization in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 8, pp. 1688-1701, December 1999.

[40] J. Liu and P. Moulin, "Information-theoretic analysis of interscale and intrascale dependencies between image wavelet coefficients," *IEEE Transactions on Image Processing*, vol. 10, pp. 1647-1658, November 2001.

[41] M. K. Mihcak, I. Kozintsev, K. Ramchandran, and P. Moulin, "Low-complexity image denoising based on statistical modeling of wavelet coefficients," *IEEE Signal Processing Letters*, vol. 6, pp. 300-303, December 1999.

[42] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, pp. 205-220, April 1992.

[43] "Final report from the Video Quality Expert Group on the validation of objective models of video quality assessment,Phase II" *http://www.vqeg.org*, August 2003.

[44] M. J. Tsai, J. D. Villasenor, and F. Chen, "Stack-run image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 519-521, October 1996.

[45] J. Vass, B. Chai, K. Palaniappan and X. Zhuang, "Significance-linked connected component analysis for very low bit-rate wavelet video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 630-647, June 1999.

[46] D. Lazar and A. Averbuch, "Wavelet-based video coder via bit allocation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 815-832, July 2001.

[47] K. K. Lin and R. M. Gray, "Wavelet video coding with dependent optimization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, pp. 542-553, April 2004.

[48] D. Marpe and H. L. Cycon, "Very low bit-rate video coding using wavelet-based techniques," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 85-94, February 1999.

[49] B. Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 1374-1387, December 2000.

[50] G. Minami, Z. Xiong, A. Wang, and S. Mehrotra, "3-D wavelet coding of video with arbitrary regions of support," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 1063-1068, September 2001.

[51] C. He, J. Dong, Y. F. Zheng, and Z. Gao, "Optimal 3-D coefficient tree structure for 3-D wavelet video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 961-972, October 2003.

[52] J. Y. Tham, S. Ranganath, and A. A. Kassim, "Highly scalable wavelet-based video codec for very low bit-rate environment," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 12-27, January 1998.

[53] S. Choi and J. W. Woods, "Motion-compensated 3-D subband coding of video," *IEEE Transactions on Image Processing*, vol. 8, pp. 155-167, February 1999.

[54] J. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Transactions on Image Processing*, vol. 3, pp. 559-571, September 1999.

[55] S. Mallat and W. L. Hwang, "Singularity detection and processing with wavelets," *IEEE Transactions on Information Theory*, vol. 38, pp. 617-643, March 1992.

[56] N. M. Namazi, P. Penafiel, and C. M. Fan, "Nonuniform image motion estimation using Kalman filtering, " *IEEE Transactions on Image Processing*, vol. 3, pp. 678-683, September 1994.

[57] S. A. Martucci, I. Sodagar, T. Chiang and Y. Q. Zhang, "A zerotree wavelet video coder," IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, pp. 109-118, February 1997.

[58] A. S. Lewis and G. Knowles, "Video compression using 3D wavelet transforms," *Electronics Letters*, vol. 26, pp. 396-398, March 1990.

[59] "Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 2: Video," *ISO/IEC 11172-2*, March 1993.

[60] "Information technology - Generic coding of moving pictures and associated audio information: Video," *ISO/IEC 13818-2*, March 1995.

[61] C. M. Kim, B. U. Lee, and R. H. Park, "Design of MPEG-2 video test bitstreams," *IEEE Transactions on Consumer Electronics*, vol. 45, pp. 1213-1220, November 1999.

[62] "Information technology - Coding of audio-visual objects - Part 1: Systems," *ISO/IEC 14496-1:2004*, November 2004.

[63] F. Moscheni, S. Bhattacharjee, and M. Kunt, "Spatiotemporal segmentation based on region merging," *IEEE Transactions on Pattern Analysis and Machine Intellidience*, vol. 20, pp. 897-915, September 1998.

[64] A. Neri, S. Colonnese, G. Russo, and P. Talone, "Automatic moving object and background separation," *Signal Processing*, vol. 66, no. 2, pp. 219-232, April 1998.

[65] C. Kim and J. N. Hwang, "Fast and automatic video object segmentation and tracking for content-based applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 122-129, February 2002.

[66] I. Kompatsiaris and M. G. Strintzis, "Spatio-temporal segmentation and tracking objects for visualization of video conference image sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 1388-1402, December 2000.

[67] M. Hötter and R. Thoma, "Image segmentation based on object oriented mapping parameter estimation," *Signal Processing*, vol. 15, pp. 315-334, October 1988.

[68] P. Bouthemy and E. Francois, "Motion segmentation and qualitative dynamic scene analysis from an image sequence," *International Journal on Computter Vision*, vol. 10, pp. 157-182, April 1993.

[69] R. Mech and M. Wollborn, "A noise robust method for 2-D shape estimation of moving objects in video sequences considering a moving camera," *Signal Processing*, vol. 66, pp. 203-217, April 1998.

[70] T. Meier and K. N. Ngan, "Video segmentation for content-based coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 1190-1203, December 1999.

[71] S. Y. Chien, S. Y. Ma, and L. G. Chen, "Efficient moving object segmentation algorithm using background registration technique," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 577-586, July 2002.

[72] Y. Tsaig and A. Averbuch, "Automatic segmentation of moving objects in video sequences: A region labeling approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 597-612, July 2002.

[73] P. Salembier and F. Marques, "Region-Based Representations of Image and Video: Segmentation Tools for Multimedia Services," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 1147-1169, December 1999.

[74] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, pp. 384-401, July 1985.

[75] G. D. Borshukov, G. Bozdagi, Y. Altunbasak, and A. M. Tekalp, "Motion segmentation by multistage affine classification," *IEEE Transactions on Image Processing*, vol. 6, pp. 1591-1594, November 1997.

[76] M. M. Chang, A. M. Tekalp, and M. I. Sezan, "Motion-field segmentation using adaptive MAP criterion," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. V, pp. 33-36, Minneapolis, MN, April 1993.

[77] D. W. Murray and B. F. Buxton, "Scene segmentation from visual motion using global optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 220-228, March 1987.

[78] J. Y. A. Wang and E. H. Adelson, "Representing moving images with layers," *IEEE Transactions on Image Processing*, vol. 3, pp. 625-638, September 1994.

[79] M. M. Chang, M. I. Sezan, and A. M. Tekalp, "An algorithm for simultaneous motion estimation and scene segmentation," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. V, pp. 221-224, Adelaide, Australia, April 1994.

[80] M. M. Chang, A. M. Tekalp, and M. I. Sezan, "Simultaneous motion estimation and segmentation," *IEEE Transactions on Image Processing*, vol. 6, pp. 1326-1333, September 1997.

[81] C. Stiller, "Object-based estimation of dense motion fields," *IEEE Transactions on Image Processing*, vol. 6, pp. 234-250, February 1997.

[82] E. Tuncel and L. Onural, "Utilization of the recursive shortest spanning tree algorithm for video object segmentation by 2-D affine motion modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 776-781, Auguest 2000.

[83] A. A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, and T. Sikora, "Image sequence analysis for emerging interactive multimedia sevices–The European COST 211 framework," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 802-813, November 1998.

[84] E. R. Dougherty, *Mathematical Morphology in Image Processing*, CRC Press, 1993.

[85] S. R. Sternberg, "Grayscale morphology," *Computer Vision, Graphics and Image Processing*, vol. 35, pp. 333-355, September 1996.

[86] F. Meyer and S. Beucher, "Morphological segmentation," *Journal of Visual Communication and Image Representation*, vol. 1, pp. 21-46, October 1990.

[87] P. Salembier and M. Paradas, "Hierarchical morphological segmentation for image sequence coding," *IEEE Transactions on Image Processing*, vol. 3, pp. 639-651, September 1994.

[88] M. Pardas and P. Salembier, "3-D morphological segmentation and motion estimation for image sequences," *Signal Processing*, vol. 38, pp. 31-43, September 1994.

[89] D. Wang, "Unsupervised video segmentation based on watersheds and temporal tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 539-546, September 1998.

[90] J. G. Choi, S. W. Lee, and S. D. Kim, "Spatio-temporal video segmentation using a joint similarity measure," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 279-286, April 1997.

[91] P. Salembier, P. Brigger, J. R. Casas, and M. Pardas, "Morphological operators for image and video compression," *IEEE Transactions on Image Processing*, vol. 5, pp. 881-898, June 1996.

[92] F. Marques and C. Molina, "An object tracking for content-based functionalities," in *Proceedings of SPIE Visual Communications Image Processing*, vol. 3024, pp. 190-199, San Jose, CA, Feburary 1997.

[93] Y. Zhong, A. K. Jain, and M. P. Dubuisson-Jolley, "Object tracking using deformable templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 544-549, May 2000.

[94] N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 266-280, March 2000.

[95] T. Meier and K. N. Ngan, "Automatic segmentation of moving objects for video object plane generation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 525-538, September 1998.

[96] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 850-863, September 1993.

[97] Y. Liu and Y. F. Zheng, "Video Object Segmentation and Tracking Using y-Learning Classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, pp. 885-899, July 2005.

[98] P. Gerken, "Object-based analysis-synthesis coding of image sequences at very low bit rates," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, pp. 228-235, June 1994.

[99] E. Francois, J. F. Vial, and B. Chupeau, "Coding algorithm with regionbased motion compensation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 97-108, Feburary 1997.

[100] B. Talluri, K. Oehler, T. Bannon, J. D. Curtney, A. Das, and J. Liao, "A robust, scalable, object-based video compression technique for very low bit-rate coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 221-233, Feburary 1997.

[101] D. S. Cho and R. H. Park, "An object-oriented coder using block-based motion vectors and residual image compensation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 316-327, June 1998.

[102] G. Calvagno, R. Rinaldo, and L. Sbaiz, "Three-dimensional motion estimation of objects for video coding," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 86-97, January 1998.

[103] S. C. Cheng, "Visual pattern matching in motion estimation for object-based very low bit-rate coding using moment-preserving edge detection," *IEEE Transactions on Multimedia*, vol. 7, pp. 189-200, April 2005.

[104] G. Qian, R. Chellappa, and Q Zheng, "Bayesian algorithms for simultaneous structure from motion estimation of multiple independently moving objects," *IEEE Transactions on Image Processing*, vol. 14, pp. 94-109, January 2005.

[105] M. Gilge, T. Engelhart, and R. Mehlan, "Coding of arbitrarily shaped image segments based on a generalized orthogonal transform," *Signal Processing: Image Communication*, vol. 1, pp. 153-180, 1989.

[106] T. Sikora, "Trends and perspectives in image and video coding," in *Proceedings of the IEEE*, vol. 93, pp. 6-17, 2005.

[107] Z. Wu and T. Kanamaru, "Block-based DCT and wavelet selective coding for arbitrarily shaped images," in *Proceedings of Visual Communication and Image Processing*, pp. 658-665, San Jose, CA, January 1997.

[108] J. H. Kim, J. Y. Lee, E. S. Kang, and S. J. Ko, "Region-based wavelet transform for image compression," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 45, pp. 1137-1140, Auguest 1998.

[109] T. Sikora and B. Makai, "Shape-adaptive DCT for generic coding of video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, pp. 59-62, Feburary 1995.

[110] P. Kauff, B. Makai, S. Rauthenberg, U. Golz, J. Lameillieure, and T. Sikora, "Functional coding of video using a shape-adaptive DCT algorithm and an object-based motion compensation toolbox," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 181-196, Feburary 1997.

[111] H. Barnard, J. Weber, and J. Biemond, "A region-based discrete wavelet transform for image coding," in *Wavelets In Image Communication*, M. Barlaud, Elsevier, Amsterdam, the Netherlands, 1994.

[112] S. Li and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 725-743, August 2000.

[113] J. Li and S. Lei, "Abitrary shape wavelet transform with phase alignment," in *Proceedings of International Conference on Image Processing*, vol. 4, pp. 683-687, Chicago, IL, October 1998.

[114] A. Said and W. A. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243-250, June 1996.

[115] G. Minami, Z. Xiong, A. Wang, and S. Mehrotra, "3-D wavelet coding of video with arbitrary regions of support," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 1063-1068, September 2001.

[116] J. E. Fowler, "Shape-adaptive coding using binary set splitting with k-d trees," in *Proceedings of International Conference on Image Processing*, pp. 1301-1304, Singapore, October 2004.

[117] E. J. Balster and Y. F. Zheng, "Virtual-object video compression," in *IEEE 48th Midwest Symposium on Circuits and Systems*, vol. 2, pp. 1700-1704, August 2005.

[118] G. Xing, J. Li, S. Li, and Y. Q. Zhang, "Arbitrarily shaped video-object coding by wavelet," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 1135-1139, October 2001.