# 國立交通大學

## 電子工程學系 電子研究所碩士班

## 碩士論文

# 用於畫面之間的小波轉換編碼以人類視覺系統為基礎的位元控制法

# HVS-based Rate Control Algorithm for Interframe Wavelet Video Coding

研究生： 洪朝雄

指導教授： 杭學鳴 博士

中華民國 九十四 年 六 月

# 用於畫面之間的小波轉換編碼以人類視覺系統為基礎的位元控制法

# HVS-based Rate Control Algorithm for Interframe Wavelet Video Coding


研 究 生：洪朝雄　　Student: Chao-Hsiung Hong

指導教授：杭學鳴 博士　　Advisor: Dr. Hsueh-Ming Hang

國 立 交 通 大 學

電子工程學系 電子研究所碩士班


碩　士　論　文

A Thesis

Submitted to the Institute of Electronics

College of Electrical Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of Requirements

for the Degree of

Master of Science

in

Electronics Engineering

June 2005

Hsinchu, Taiwan, Republic of China

中華民國 九十四 年 六 月

# 用於畫面之間的小波轉換編碼以人類視覺系統為基礎的位元控制法

研究生：洪朝雄　　指導教授：杭學鳴

國立交通大學 電子工程學系 電子研究所碩士班

## 摘要

因為在大多數的應用中，不同的接收者會有不同的承受量，故可調整性(scalability)在今天的多媒體傳輸中是一個重要的特性。用於畫面之間的小波轉換編碼(Interframe Wavelet Video Coding)是一個新的視訊編碼方式且能提供良好的可調整性。因此這個編碼方式在近年來受到不少矚目，而且已經有很多的研究和改良來增進它的效能。

在很多環境下，人眼都是視訊品質的最後判斷所在。然而，在設計視訊編碼時要包含人類視覺卻很困難。我們必須要能把客觀的“數學上的不同”轉換成主觀的“視覺上的不同”，也就是說，我們必須要把普通的“量化錯誤”轉換成“人類視覺上的加重錯誤”。

在位元控制法(rate control algorithm)中，每個在用於畫面之間的小波轉換編碼的截斷點(truncation point)都有自己相關聯的失真(distortion)和位元長度(bits length)。而每個截斷點的斜率(slope)就是把失真的差異(distortion difference)除以位元差異(bit difference)所得到的商。在最佳化理論中(optimization theory)，擁有較高斜率的截斷點有較高的優先權被傳送。在本論文中，我們提出一個方法，就是說我們把每個截斷點的斜率乘上一個由人類視覺系統算出來的比重。故這個經過視覺加重的斜率會成為位元控制法中判斷的標準。我們的模擬會指出最後的重建影像有較低的最高訊號雜訊比(PSNR)和較佳的視覺品質。

# HVS-based Rate Control Algorithm for Interframe Wavelet Video Coding

Student: Chao-Hsiung Hong Advisor: Dr. Hsueh-Ming Hang

Department of Electronics Engineering &
Institute of Electronics
National Chiao Tung University

# Abstract

Scalability is an important feature in today's multimedia transmission because in many applications receivers have very different capabilities. Interframe wavelet video coding is a new video coding algorithm that can achieve fine-scale scalability. Therefore, it has received a lot of attention recently and many research and development projects have been conducted to improve its performance.

For most entertainment purposes, human eyes are the final judge of the video quality. However, it is rather sophisticated to include the human perception in the video codec design. We need to transform the objective "mathematical difference" into the subjective "visual difference", i.e., we need to convert the ordinary "quantization error" to the "human-visual weighted error".

In the rate control algorithm, each truncation point in the interframe wavelet video coding has its associated distortion and bits length. The slope of each truncation point is the quotient of the distortion difference divided by the bit difference. Based on the optimization theory, the truncation point with a larger slope should have a higher priority to transmit. In this study, we propose a method that we weight the truncation point slope by a weighting factor, which is derived based on the human visual system. Thus, the visually-weighted slopes become the criterion in rate control. Our simulations indicate that the reconstructed frames may have lower PSNR but higher visual quality.

# 誌謝

感謝杭學鳴老師，這兩年不厭其煩地指導我，讓我漸漸學習到如何作研究以及作研究該注意的事。

再來要感謝我的家人。謝謝我父母和我哥哥總是在背後支持我和給我許多地鼓勵。

還有謝謝實驗室裡的許多的學長姐和同學，大家給了我許多研究方面的建議，還有實驗室優良的環境，讓我可以好好地從事研究工作。

經過了兩年的研究生活，雖然中間碰過不少困難，不過感謝大家的幫忙，讓我能完成我的學業。在此把這本論文獻給所有幫助過我的人和我所感謝的人。

# Contents

# List of Figures

# List of Tables



# Chapter 1
# Introduction

Digital video compression technology has an explosive growth in the past 20 years. The invention of digital video products, such as VCD and DVD, is due to the advances of the digital compression technology. Owing to the rapid development of the internet transmission, it is also important to transmit the video data through the network. Due to the different network bandwidth and different receiver storage capacity, many methods have been investigated to solve the problem of transmitting the compressed video bitstream through the internet. The concept of "scalability" is one of the methods that solve this problem. The "scalability" means that the bitstream can be truncated and decoded anywhere on the bitstream; thus, we can generate the bitstream only once then truncate it to meet the requirements.

However, in a traditional scalable video system, because of the lower compression efficiency and course-step in scalability (typically, 2 or 3 layers), its adaptation is not yet so popular. The new technique of fine-granularity scalability is introduced recently [1]. Ohm proposed a motion-compensated t+2D frequency coding structure [2]. This coding structure is suitable for scalable video coding with many fine steps. Woods proposed a coding technique called "interframe wavelet video coding" [3]. This coding technique can offer fine-granularity SNR, temporal and spatial scalability at the same time, while it still maintains acceptable compression efficiency.

The main concept of interframe wavelet video coding is subband coding. It removes the temporal redundancy by using the motion-compensated (wavelet) filtering technique along the temporal axis. Then it uses the spatial wavelet

decomposition to the temporal wavelet-filtered output frames. Then we can use the bit-plane coding scheme to code wavelet coefficients and calculate the slope of each fractional bit-plane truncation point to achieve optimal rate control. By this rate control scheme, we can achieve fine-granularity scalability [4].

The quality measure that often be used to determine the quality of images is PSNR. But human eyes have different sensitivity on different regions and frequency bands, the image that has high PSNR value may not have high visual quality. Human eyes usually have higher sensitivity on the low frequency bands and lower sensitivity on the high frequency bands. For the different region, human eyes usually have higher sensitivity in the flat region than in the texture region. We can incorporate human visual system (HVS) to encode each subband to achieve higher visual quality.

In this research, we propose a rate control algorithm based on HVS to achieve high visual quality. We apply HVS on spatial frequency and luminance component. There two weighting factors, intra-subband weighting factor and inter-subband weighting factor, that we found will be introduced. The final reconstructed images will have higher visual quality, especially in large flat region. The PSNR of final reconstructed images will be lower. In the future, we will extend this algorithm to temporal frequency and chrominance component.

The thesis is organized as follows. In Chapter 2, we will introduce the basic concept and the scalability of scalable video coding. Then we will introduce the program we used in Chapter 3. We will introduce some basic idea of HVS in Chapter 4. The algorithm we developed is introduced in Chapter 5 and Chapter 6 is the conclusion and future work.

# Chapter 2
# Scalable Video Coding

## 2.1 Introduction

Digital Video is now very popular in our daily life. For example, DVD and VCD are all digital video. If the digital video has high quality, it usually has a large amount of data. So it needs large bandwidth to transmit or large space to store. To solve this problem, we need to compress the digital video in order to make its data size smaller. Digital video compression technique has been developed in the past three decades and much research has been done to analyze the digital video sequences. Several video standards have been developed, for example, MPEG-2, and H.261. Based on different theoretical foundations, we can classify the video coding into two groups as shown in Figure 2-1.

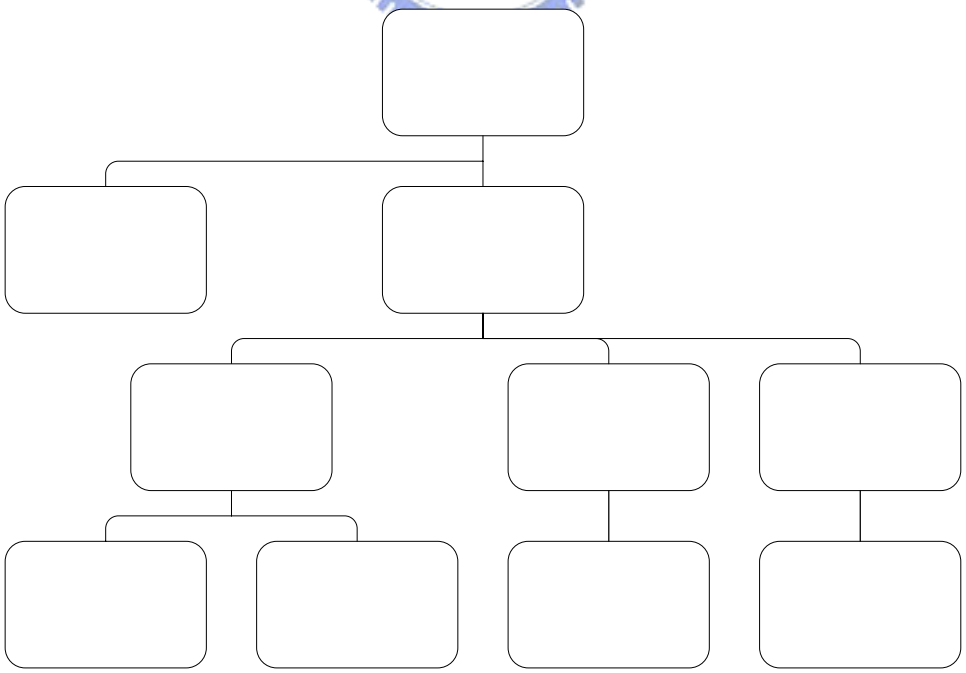


Figure 2-1 Classifications of video coders.

From Figure 2-1, we can see that video coders can be classified into “model based”

and “signal based” two groups. If the video coding algorithm is based on object modeling and analysis of object parameters, it belongs to model based video coding. Model based video coding algorithms usually need a profound analysis of the video contents and are quite complicated. Because of inefficiency and complexity of video object content analysis, model based video coding algorithms are often not so popular.

On the other hand, the signal based video coding algorithms consider the objects as the combination of the set of basic signals. So they often use filters to decompose the video sequences into different basic signals. The signal decomposition of these algorithms has two spatial dimensions (horizontal and vertical) and one temporal dimension. Both spatial and temporal decomposition are used to remove in-between redundancies. We usually use discrete cosine transform (DCT) or discrete wavelet transform (DWT) to do spatial decomposition and motion compensated temporal filtering (MCTF) or motion compensated prediction (MCP) to do temporal decomposition.

The motion compensated approaches decompose the source output into different frequency subband using block transforms. But decomposition of the source output into blocks will generate coding artifacts at the block edges called blocking effects. Another approach, which can avoid this blocking artifact, is the subband video coding. The subband video coding transforms the total frame into different subbands in spatial and temporal domain, so it can remove blocking effect.

## 2.2 Subband Video Coding

Subband video coding uses subband filters to remove the spatial and temporal redundancies of the video sequences. Generally speaking, the behavior of the spatial and temporal signals of a video sequence is quite different. For temporal signal, if something moves fast in the video sequence, then the video sequence has high

frequency temporal signal component. The spatial signal will be only considered in the still image. If a still image has many edges or different luminance component in a small area, then it has high frequency spatial signal component. Spatial signal has 2 dimensions (horizontal and vertical) and temporal signal has 1 dimension, so the decomposition of spatial signal is often done twice. Typical 3-D subband signal decomposition is shown in Figure 2-2.

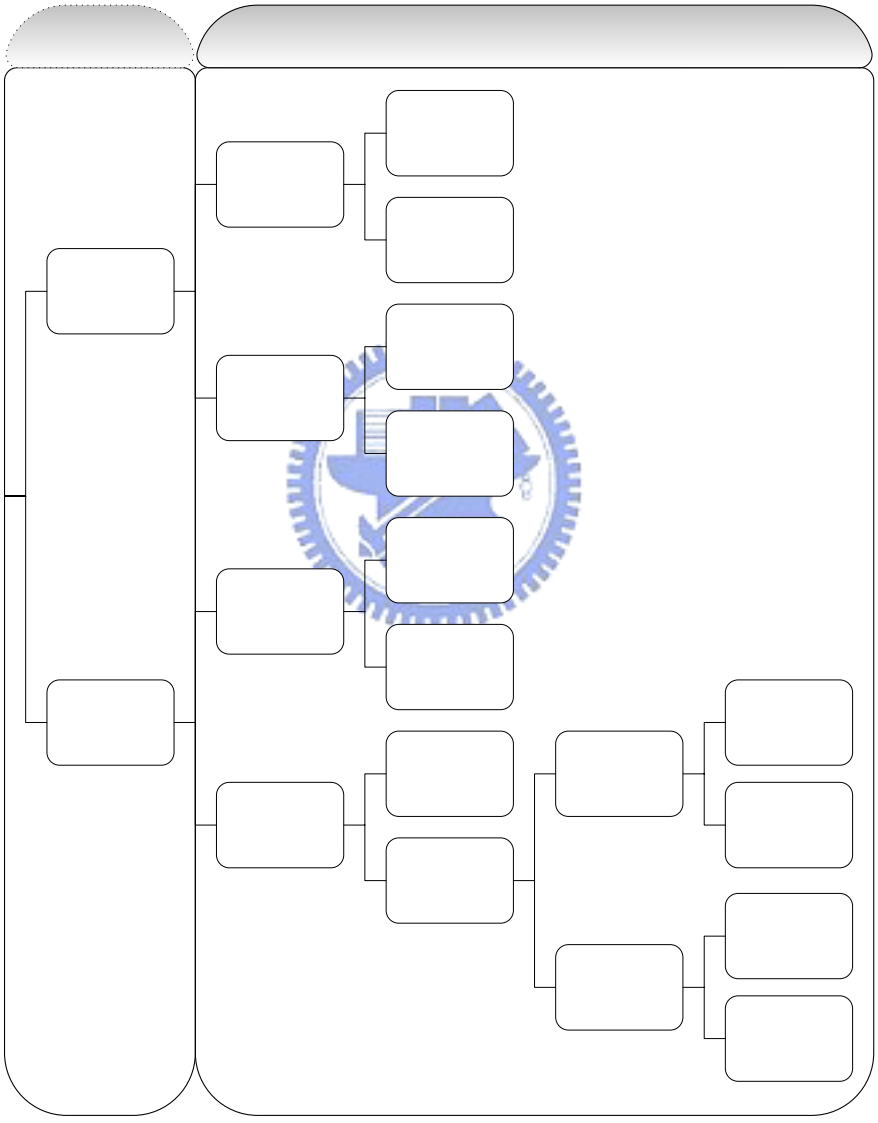


Figure 2-2 Typical 3-D subband decomposition.

After spatial and temporal decomposition, the data are sent to quantize and coding. After coding, the coded data is packaged and transmitted to the receiver to decode.

**Temporal Subban**
**Decomposition**

Because human eyes have different sensitivity on different frequency subbands, we can quantize the frequency subbands with higher sensitivity by smaller step size and other frequency subbands by larger step size. Thus we can get the reconstructed video sequence with higher visual quality but lower PSNR value. We will introduce the temporal decomposition and spatial decomposition in next two subsections.

### 2.2.1 Temporal Subband Decomposition

Temporal subband decomposition can be simply done by use a low pass filter and a high pass filter along the temporal axis. The filter used more often is Haar filter. But the result is not usually good because the energy is not compacted very well. The result is shown in Figure 2-3. We can see that the output of the low-pass filter would be a blurred image, a moving average of the original video sequence, and the output of the high-pass would be the difference of the original video sequence.

Figure 2-3 The temporal filtered images using Haar filter (left: low pass, right: high pass).

Kronander used motion compensated technique to solve this problem [5]. For two consecutive frames, we use forward block motion estimation first and backward block motion estimation second. The forward motion compensated reconstructed frame is then used to do temporal filtering with the second frame to generate subband image. Then the backward motion compensated reconstructed frame is used to do temporal filtering with the first frame. The result frames has better energy compaction as shown

in Figure 2-4. There may be a mismatch between these two vectors and it will cause the spatial inhomogeneity. The mismatch often occurs in the covered and uncovered area on the frame, as shown in Figure 2-5

Figure 2-4 Temporal filtering with motion compensation (left: low pass, right: high pass).

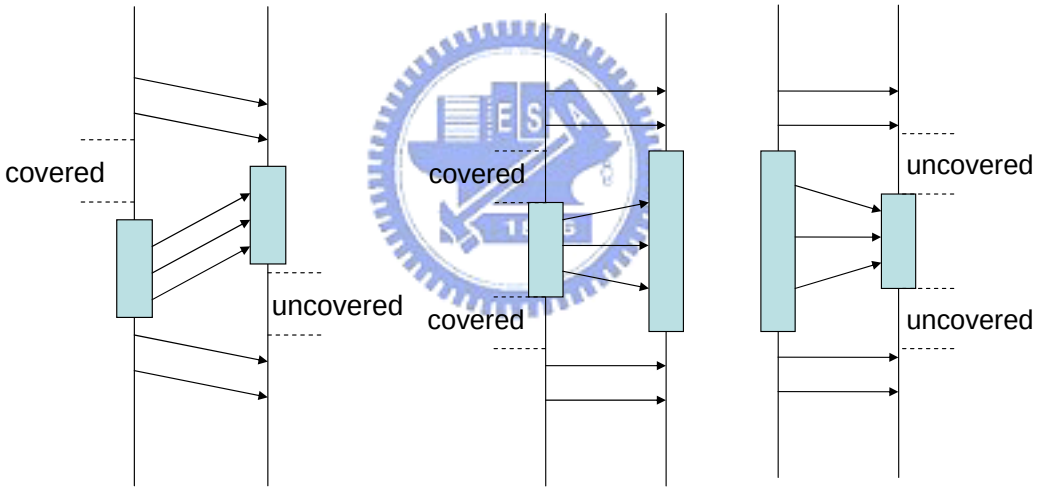



Figure 2-5 Vector mismatch caused by moving and zooming objects.

Ohm proposed a method to solve the spatial inhomogeneity [2]. He showed that it is possible to overcome the mismatch of motion trajectories by using the concept of connected and unconnected pixels. In the proposed algorithm, each pixel is classified as covered, uncovered or connected by using the information derived from the motion vector map. Then Haar filter is used to do temporal filtering to find the high-pass coefficients and the low-pass coefficients. If the integer pixel accuracy motion estimation is used, this method can achieve perfect reconstruction.

Hsiang and Woods proposed an invertible half-pixel motion estimation three-dimensional analysis/synthesis system for video coding [6]. If we assume that *dx* and *dy* are the horizontal and vertical displacement vector between previous and current frames, and they can be pixel or half pixel. Then we can classify the motion compensated blocks into four different kinds, as shown in Figure 2-6. The motion compensated blocks would map to different location of the image, but would lie in horizontal, vertical, diagonal, or overlapped position. Therefore, temporal Haar filtering can be done along those directions to achieve half-pixel-accurate motion estimation.

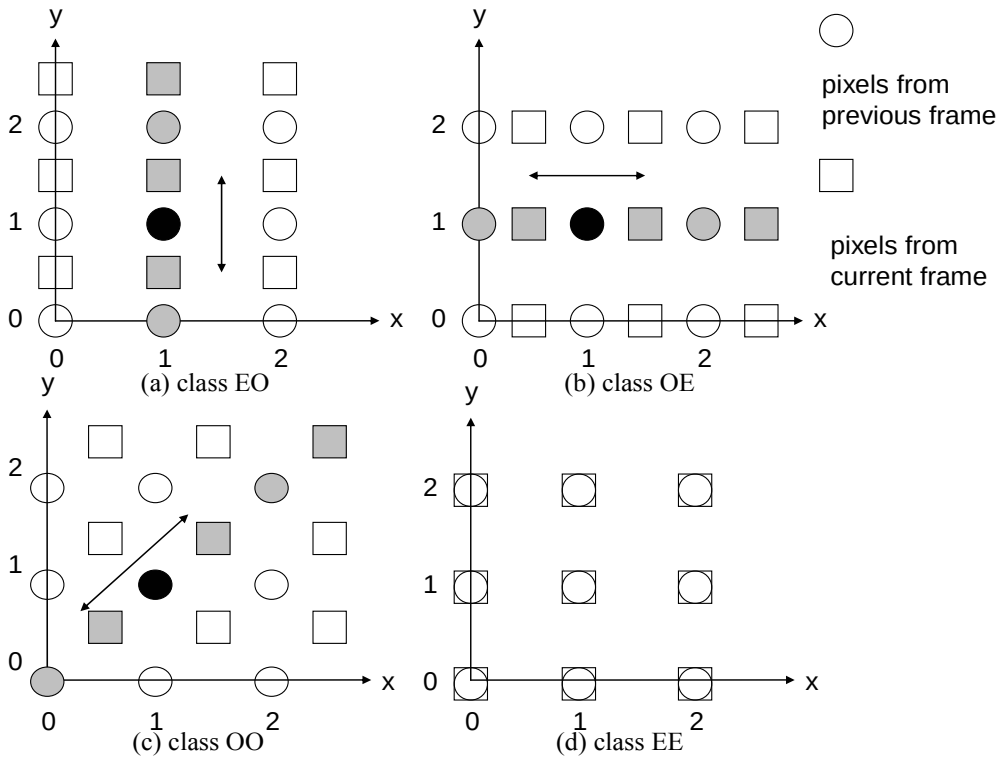



Figure 2-6 The spatial lattices of two consecutive frames after motion estimation. The black circle is the pixel being processed. The gray pixels and arrows indicate the direction of filtering. (a)class EO:2*dx* even and 2*dy* odd, (b)class OE: 2*dx* odd and 2*dy* even, (c)class OO: 2*dx* odd and 2*dy* odd, (d)class EE: 2*dx* even and 2*dy* even.

Pesquet-Popescu and Bottreau proposed a lifting scheme to do temporal filtering [7]. By separating the process of deducting the low-pass and high-pass frequencies, interpolation filters can be used without interfering with the filtering process.

Temporal filtering techniques are still being researched and developed. The goal of the temporal filter is make the energy compacted well.

### 2.2.2 Spatial Subband Decomposition

Spatial decomposition is done along horizontal and vertical directions. The still image is separated into the spatial subband then each subband is encoded independently. The image is reconstructed from the low subband data to high subband data. The major differences are how to choose the analysis and synthesis filters. In other words, that is how to choose the decomposition signal. The performance of the filter will affect the quality of the reconstructed images.

The most popular spatial subband decomposition is the wavelet transform. Wavelet transform is a type of localized time-frequency analysis; therefore, the transform coefficients reflect the energy distribution of the source signal in both space and frequency domains. Figure 2- 7 shows an example of the spatial decomposition.

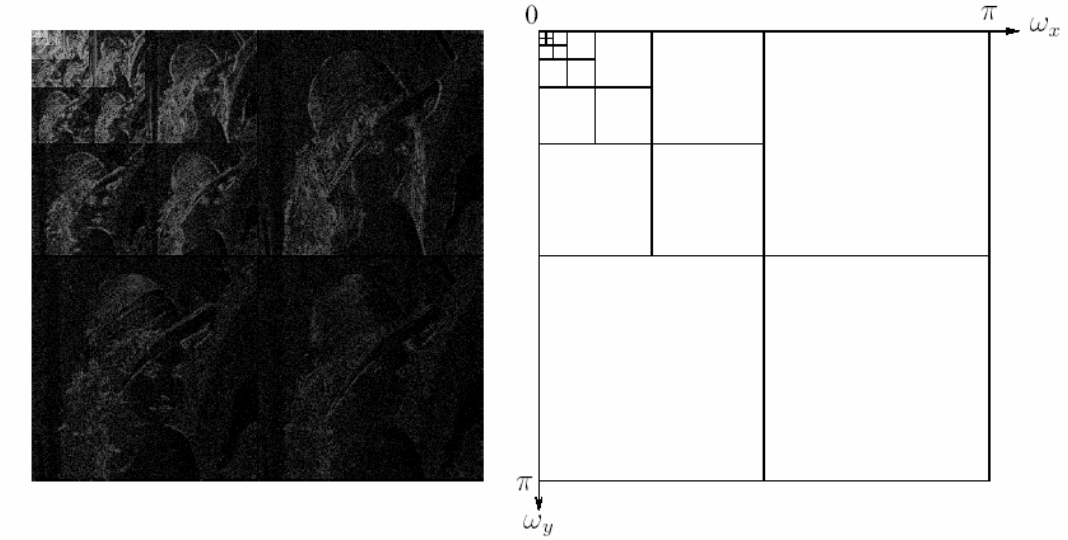



Figure 2- 7 Spatial decomposition (left: transformed image, right: frequency partion).

### 2.2.3 Coding

Shapiro proposed a coding algorithm called “Embedded image coding using zerotrees of wavelet coefficients (EZW)” [8]. It is a simple but effective coding structure. It arranges the coded data in the order of importance so it is suitable for progressive transmission.

Taubman proposed a coding algorithm called “Embedded Block Coding with Optimized Truncation of the embedded bit-stream (EBCOT)” [9]. This is a coding algorithm that JPEG2000 used. It is a fractional bit plane coding and can match the requirement of the rate control.

Woods proposed a coding algorithm called “Embedded Zero Block Coding (EZBC)” [10]. It combines the advantages of the zero-tree/-block coding and context modeling of the subband/wavelet coefficients.

## 2.3 Interframe Wavelet Video Coding

### 2.3.1 Introduction

The efficient family of interframe wavelet video codecs was proposed by Woods and his coworkers [6] [10] [12] and can achieve rate/SNR, spatial, and temporal scalability. It was first presented by Woods et al for the MPEG digital cinema encoding tool [11]. Many research and development have been made to improve the performance of the interframe wavelet video coder today. In the rest of this thesis, if not specifically stated, the interframe wavelet coding algorithm referred is the latest version proposed by Woods and Chen [12].

The interframe wavelet coder is one kind of motion compensated 3-D subband coder. 3-D is 2 spatial dimensions (horizontal and vertical) and 1 temporal dimension. This coding algorithm is also known as the “Motion Compensated Temporal

Filtering – Embedded Zero Block Coding (MCTF-EZBC or MC-EZBC)". This coding algorithm uses motion compensated temporal filtering techniques when doing temporal subband decomposition. After temporal subband decomposition, each produced frame is spatially subband decomposed by wavelet transform. After temporal and spatial decomposition, the wavelet coefficient is coded by embedded zeroblock coding techniques [10]. Then we can package and truncate the coded bitstream and transmit it to the receiver and decode. The architecture of the interframe wavelet video coder is shown in Figure 2-8.

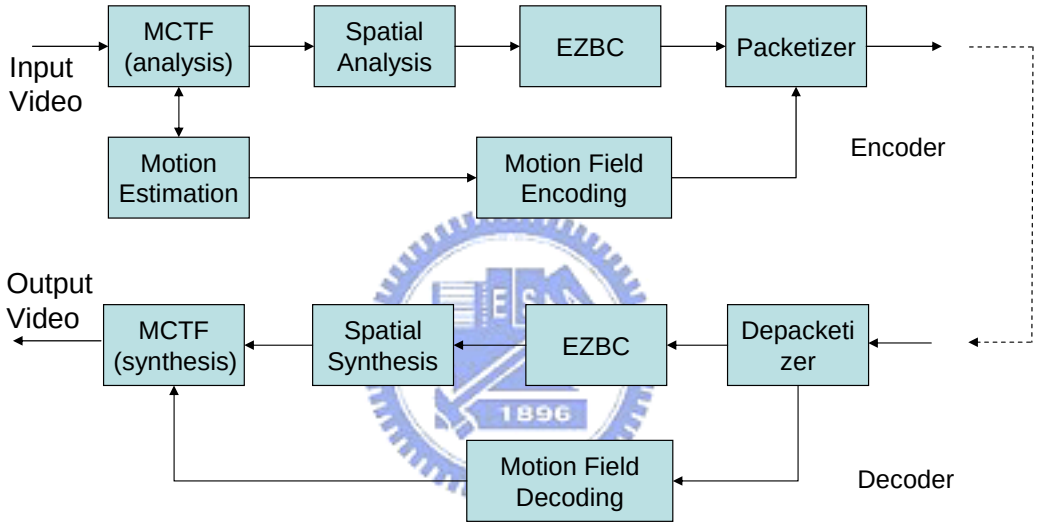



Figure 2-8 The interframe wavelet video coder.

The processing unit is GOP (group of pictures) and the frame number of each GOP is $2^n$, where n is the number of levels of temporal subband decompositions that are done on the GOP. When doing temporal subband decomposition, the motion vector map between two consecutive frames is first constructed. Then motion compensated temporal filtering is applied to the two consecutive frames to generate the temporal high-pass frame and the temporal low-pass frame.

After first temporal decomposition, the GOP would contain $2^{n-1}$ temporal high-pass frames and $2^{n-1}$ temporal low-pass frames. Then $2^{n-1}$ temporal low-pass frames would be collected to do temporal decomposition again. These $2^{n-1}$ temporal low-pass frames

would transform to $2^{n-2}$ temporal high-pass frames and $2^{n-2}$ temporal low-pass frames. The temporal decomposition is iteratively done until there is only one temporal low-pass frame. After finishing temporal decomposition, we can get a temporal filtering pyramid as shown in Figure 2-9.

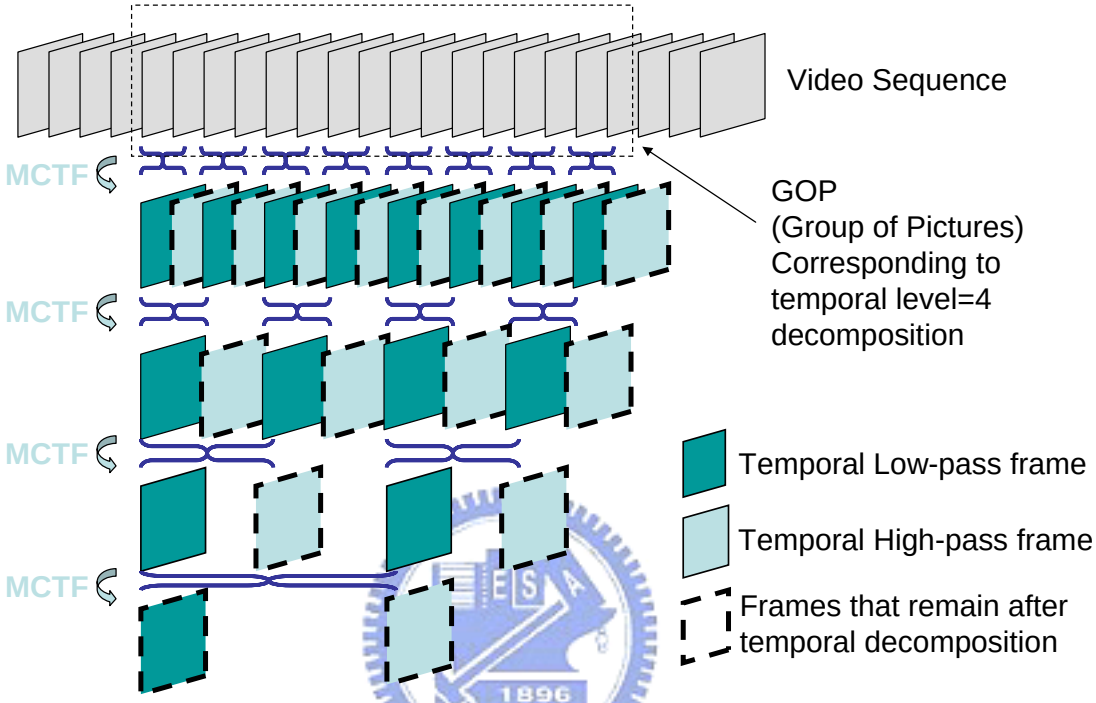



Figure 2-9 Temporal filtering pyramid.

After temporal decomposition is done, the $2^n$ frame GOP would contain $2^{n-1}$ temporal high-pass frames and one temporal low-pass frame. These frames are called residual frames. The spatial subband decomposition is then applied to each frame to create wavelet coefficients of each spatial subband. The wavelet coefficients is then coded by embedded zeroblock coding method, and then entropy-coded by arithmetic coding with context modeling [10].

## 2.3.2 Motion Compensation Temporal Filtering

The interframe wavelet video coding uses motion compensated temporal filtering (MCTF) to do temporal subband decomposition and the goal of motion compensated temporal filtering is to compact the video sequence temporal energy.

The first step of MCTF is motion estimation and there two things need to do in this step. First is using “hierarchical variable size block matching (HVSBM)” to do backward motion estimation. Second is detecting covered and uncovered pixels based on the backward motion field [13].

### 2.3.2.1 Hierarchical Variable Block Size Matching

HVSBM is a hierarchical motion estimation scheme that can reduce computational complexity and generate smooth motion vector fields. HVSBM can create better motion estimation because of its variable block size. The motion compensated temporal filtering performance depends on how well the motion trajectory, which is constructed by the motion search, matches the moving objects in the video sequence.

The motion vectors are first searched in the 64-by-64 size block. Then the block is split into four 32-by-32 subblocks. Motion vectors for subblocks are generated by refining the motion vector of the original block. This spawning process continues until the size of the subblock is 4-by-4. Figure 2-10 shows a 3 level HVSBM. Consequently, longer-range interaction is enforced at lower resolution (higher scale) levels, while shorter-range interaction is recovered at higher resolution (lower scale) levels. Finally we can get a five level motion vector quad-tree with one 64-by-64 block size motion vector at the top and 256 4-by-4 block size motion vectors at the bottom [13].

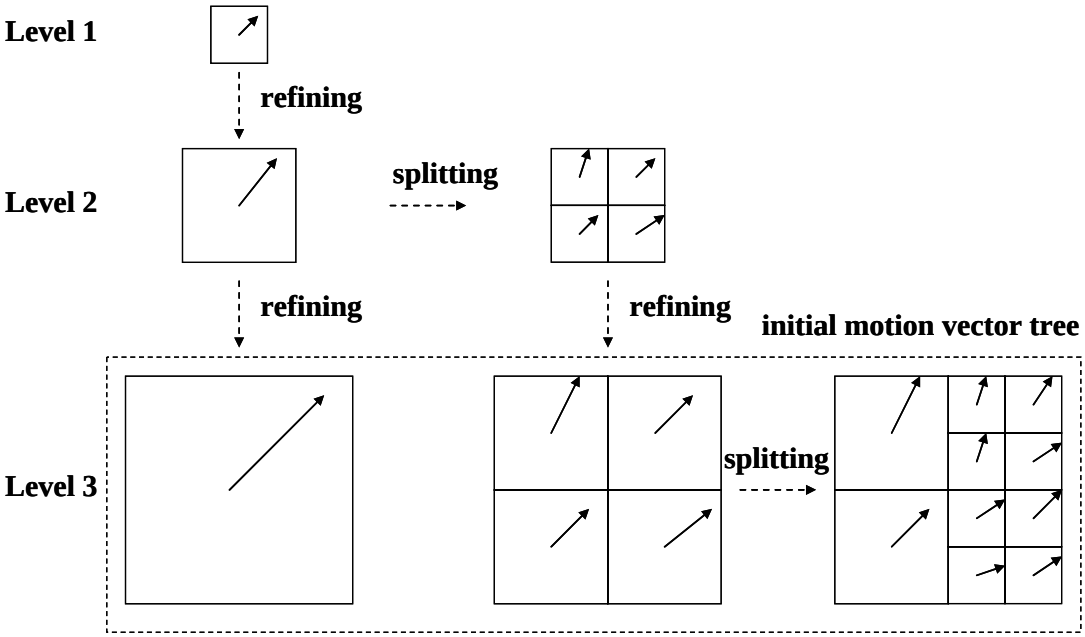



Figure 2-10 A 3 level HVSBM showing 3 subband levels [13].

## 2.3.2.2 Detection of Covered and Uncovered Pixels

There two reasons that this process is needed. One is that the motion estimation process may not be perfect because of the wrong motion trajectory; the other is that the temporal filtering is applied along the motion trajectory, so it depends on the linked condition of the pixel.

By the motion vectors we get from HVSBM, we can link every pixel in the predicted frame to another pixel in the reference frame. We can classify the linked condition of pixels on the predicted frame and reference frame. From Figure 2-11, we can see there 3 types of pixels in the reference frame and 3 types of pixels in the predicted frames.

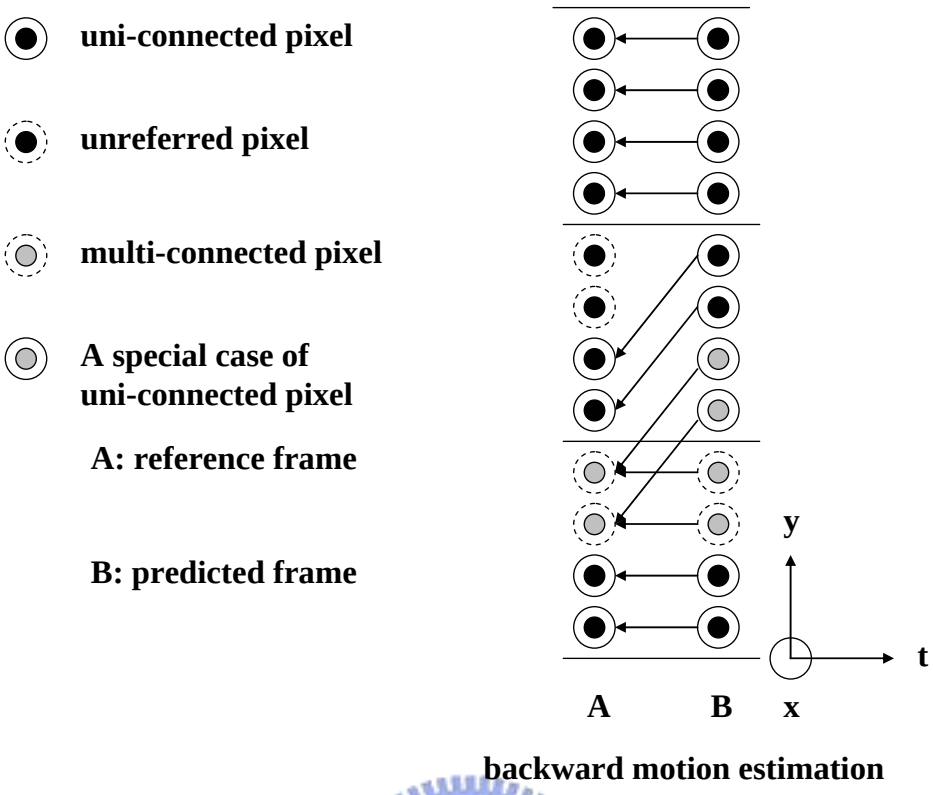



Figure 2-11 State of connection of each pixel [13].

The 3 pixel types in the reference frame are: 1) uni-connected pixel, a pixel which is used as reference by only one pixel in the predicted frame, 2) unreferred pixel, a pixel which is not reference by any pixels in the predicted frame, 3) multi-connected pixel, a pixel which is used as reference by more than one pixel in the predicted frame.

The 3 pixel types in the predicted frame are: 1) first type of uni-connected pixel, a pixel whose reference pixel in the reference frame is uni-connected pixel, 2) second type of uni-connected pixel, a pixel whose reference pixel in the reference frame is multi-connected pixels that has better response to sum of absolute difference (SAD), 3) multi-connected pixel, the rest of the pixels in the predicted frame.

Forward motion estimation is done if there are more than half of the pixels are classified as multi-connected pixels in a block of the predicted frames. If motion estimation in this direction has smaller SAD error, we call this block is an uncovered

block and pixels in this block are said to be uncovered pixels. The unreferred pixels in the reference frame are marked as covered pixels, while the rest are connected. If the number of the unconnected pixels exceeding a threshold, the interframe wavelet video coder would assume that there is a scene-change in the video sequence and it would stop temporal filtering across the two frame pairs. Otherwise, when the test fails, and temporal filtering is done, all the pixels in the predicted frame are remarked as connected pixels.

#### 2.3.2.3 Motion Vector Pruning

The motion vector pruning process is done to delete unnecessary nodes from the quad-tree created by HVSBM. For the quad-tree, each node contains the estimated motion vector for that corresponding block. The motion vector pruning process initially generated motion vector bit estimation of each node. Then the difference of the bits used for the parent and child and difference of the SAD of the parent and child are calculated. Using these two parameters as the rate and distortion measure, the rate-distortion cost of every node is generated. An iterative loop is then done to prune the leaf nodes with the highest cost until a desired rate-distortion cost is achieved.

#### 2.3.2.4 Temporal Filtering

The interframe wavelet video coding uses the lifting scheme in temporal filtering [7], which can achieve perfect reconstruction even when sub-pixel motion estimation is used. Figure 2-12 shows the lifting scheme in temporal filtering and Figure 2-13 shows the detection of connected and unconnected pixels.

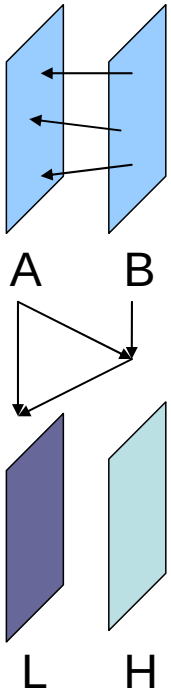



Figure 2-12 Lifting scheme in temporal filtering.

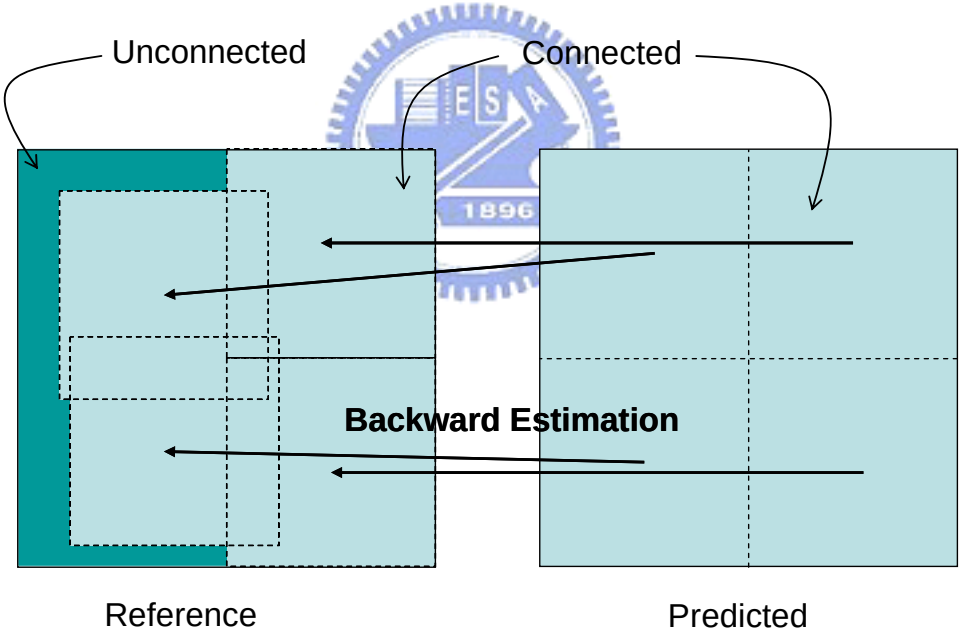



Figure 2-13 Detection of connected and unconnected pixels.

Assume that $A$ and $B$ are reference frame and predicted frame, and $\widetilde{A}$ is the interpolated frame of $A$. The notation [$m$, $n$] represents the pixel coordinates and $(d_m, d_n)$ is the motion displacement from the predicted frame $B$ points to a sub-pixel position in the reference frame $A$. In other words, $B[m, n]$ is connected to $A[m-\bar{d}_m$,

$n-\bar{d}_n$ ] where $\bar{d}_m$ and $\bar{d}_n$ are the closet integer to $d_m$ and $d_n$.

The temporal high-pass coefficients are calculated on the predicted frame by (1). The motion estimation would link every pixel in the predicted frame to another pixel in the reference frame; therefore, all pixels in the predicted frame are connected.

$$H[m,n] = \left(B[m,n] - \tilde{A}[m-d_m, n-d_n]\right)/\sqrt{2} \quad (1)$$

The temporal low-pass coefficients are generated on the reference frame and the pixels on the reference frame can be classified as connected and unconnected. The low-pass coefficients of the connected pixel are calculated by:

$$L[m-\bar{d}_m, n-\bar{d}_n] = \tilde{H}[m-\bar{d}_m+d_m, n-\bar{d}_n+d_n] + \sqrt{2}A[m-\bar{d}_m, n-\bar{d}_n] \quad (2)$$

The low-pass coefficients of the unconnected pixels are calculated by:

$$L[m,n] = \sqrt{2}A[m,n] \quad (3)$$

When decoding, $A$ can be reconstructed by:

$$A[m-\bar{d}_m, n-\bar{d}_n] = \left(L[m-\bar{d}_m, n-\bar{d}_n] - \tilde{H}[m-\bar{d}_m+d_m, n-\bar{d}_n+d_n]\right)/\sqrt{2} \quad (4)$$

After reconstruction of A, we can reconstruct B by:

$$B[m,n] = \sqrt{2}H[m,n] + \tilde{A}[m-d_m, n-d_n] \quad (5)$$

In (4), we can see $L$ and $H$ are still necessary for the reconstruction of $A$, and $H$ only contains the information of interpolated pixels in $A$. But this interpolated information is also available in $L$. So it is canceled out in (4). Thus the interpolation algorithm has no influence on the perfect reconstruction [13].

## 2.3.3 Spatial Analysis

Spatial wavelet transform is performed on every residual frame after motion compensated temporal filtering. If the video sequence is composed of YUV component, then spatial wavelet transform is performed on all the three components.

The coefficients of the used filters are shown in Table 2-1.

| index | low pass filter | high pass filter |
|---|---|---|
| 0 | 0.852699 | 0.788485 |
| ±1 | 0.377403 | -0.418092 |
| ±2 | -0.110624 | -0.040690 |
| ±3 | -0.023849 | 0.064539 |
| ±4 | 0.037829 | |

Table 2-1 The coefficients of filters.

## 2.3.4 Embedded ZeroBlock Coding

After temporal and spatial decomposition, the coefficients are coded by "Embedded ZeroBlock Coding (EZBC)". Because of the zeroblock coding and context modeling, this coding algorithm can achieve low computational complexity and high compression efficiency.

The coding process begins with the creation of the quad-tree based set partitioning data representations on bit-planes for each individual subband. The bottom of the quad-tree level is the pixel level and consists of the magnitude of each subband coefficients. Each quad-tree node of the next higher level is then set to the maximum value of its four corresponding nodes at the current level, as illustrated in Figure 2-14(a). By recursively grouping the coefficients, the top quad-tree node would correspond to the maximum magnitude of all the coefficients from the same subband.

Then the bitplanes of subband coefficients from the most significant bit toward the significant bit is progressively encoded. If a node is significant, it is split into four descendent nodes. This procedure is recursively down until the bottom level, as illustrated in Figure 2-14 (b). Once a pixel is significant, its sign bit is coded. Each bitplane coding pass is finished with a bitplane refinement subpass which further

refines the significant subband coefficients from the previous bitplane pass. So we can send data in the order of their importance in this way.

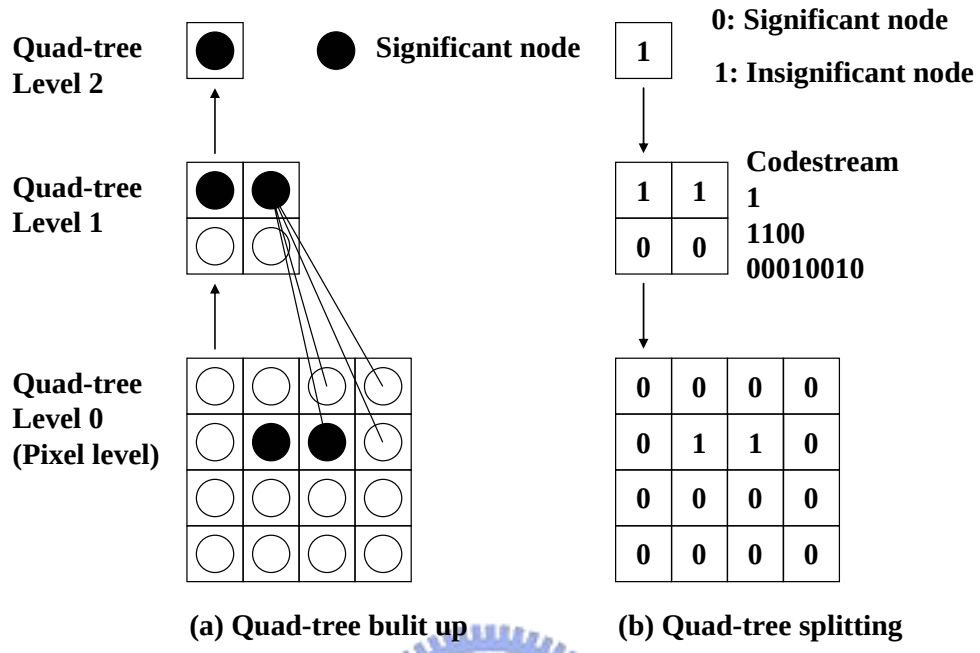



Figure 2-14 Quad-tree generation of the image [10].

## 2.3.5 Entropy Coding

This is the final process of encoding. At this time, the processed data contains motion information and the EZBC coded image data. Because HVSBM uses variable block size when doing motion estimation, the information of how the block sizes are arranged need to be coded. This information is contained in the quad-tree structure and coded in the map representation as shown in Figure 2-15. Then the map code is inserted in the encoded bit-stream.

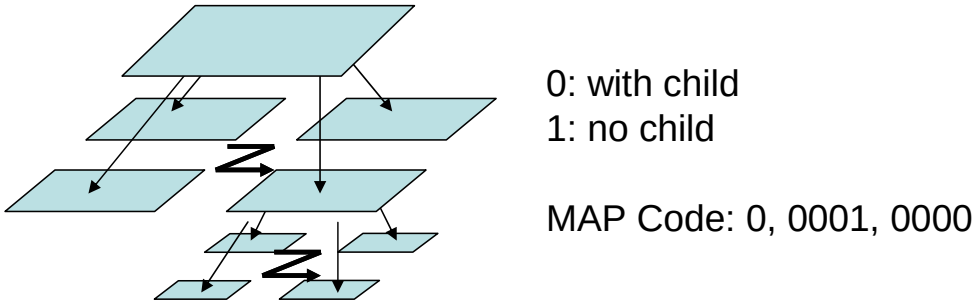



Figure 2-15 Map representation of the quad-tree.

The motion vectors of the leaf node blocks are sent into the adaptive arithmetic coder following the raster scan shown in Figure 2-16. The scanning order is the recursive raster scan of the leaf nodes in the motion vector quad-tree.

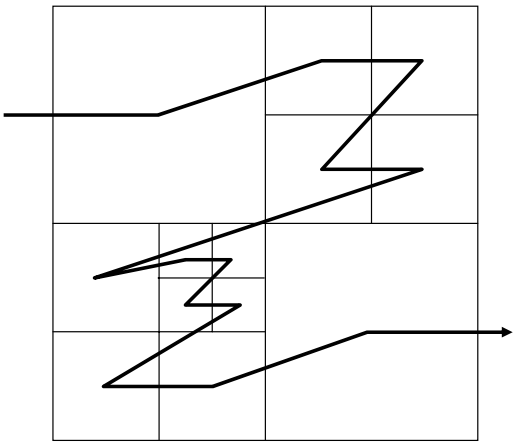


Figure 2-16 Motion vector coding scanning trail.

The arithmetic coder initially sets the probability of all symbols to the same value. Then the symbol probability is updated after each symbol is encoded by accumulating the occurrence of the symbol during encoding. Combined with EZBC's quad-tree representation of the image data, the strong statistical dependencies among bit-planes, resolution scale, and quad-tree levels are exploited [10].

## 2.4 Scalable Video Coding

Mass audiences have different viewing requirement and the bandwidth and capacity of each server on the network is different. So it is important to meet different requirement. The principle idea of the scalable video coding is that the encoded bitstream can be flexibly truncated to meet the requirements after the compressed bitstream has been generated.

Digital video can have many specifications, such as picture size, picture quality, and picture playback rate. Because different user may have different requirement on these specifications, the ability to scale and choose different combinations of these

video specifications is crucial for simultaneous distribution to disparate clients .The main concept of the scalable video coding is “generate-once, scale-many”.

Most people have more demand on the picture quality. There are three video scaling parameters that influence the viewing quality most: 1) the distortion of the picture, 2) the spatial resolution of the image, 3) the temporal resolution of the video. One major feature of the interframe wavelet coding is the ability to achieve all of the three mentioned video scalable features in one single coding algorithm. We will introduce these three scaling parameters in the following subsections.

### 2.4.1 Rate/SNR Scalability

The rate/SNR scalability is the ability that a single compressed bitstream can be decoded into different coding bit-rates/quality levels.

The basic element of the interframe wavelet video coding encoded bitstream is GOP. It is composed by GOP header, the motion information, and the image data, as shown in Figure 2-17. The motion information is required to construct the motion fields that are used in the motion compensated temporal filtering so it is usually sent without any modifications. The image data is used to construct residual frames and it will be truncated to match the requested bit rate.

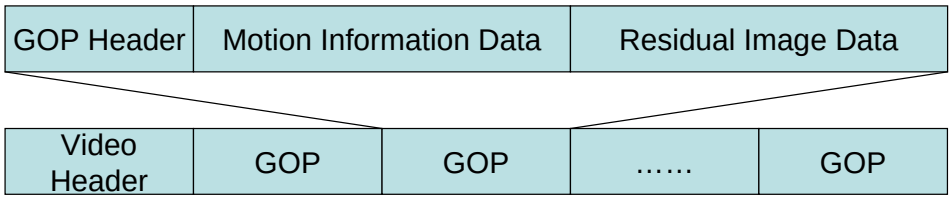



Figure 2-17 The interframe wavelet video coding encoded bitstream.

The rate/SNR scalability is achieved by truncation the image data to match the required bit rate. In the EZBC process, the bit-stream is arranged in an embedded structure such that information bits are saved in accordance to the importance of the data. During the process of the encoding of the EZBC, the information of how many

bits are used in the subband is marked as a parameter file, indicating the truncation points of the encoded residual image data in the video bit-stream.

When doing truncating, the total bit rate of the GOP header, motion information, and the image data must equal to or less than the required bit rate. So the truncating process will find the corresponding truncation point and read the image data before the truncation point then package. Figure 2-18 shows the rate/SNR scalability.

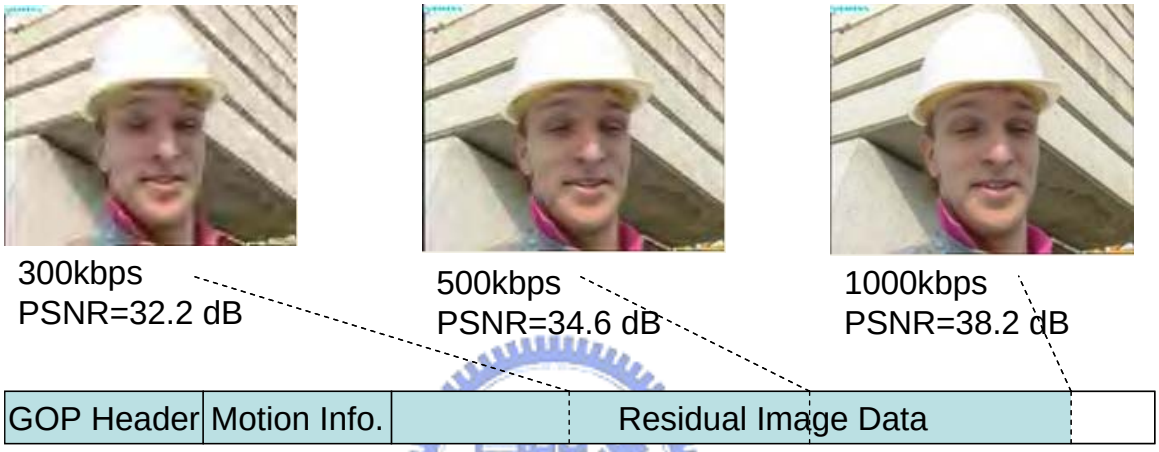



Figure 2-18 Rate/SNR scalability.

## 2.4.2 Spatial Scalability

When performing spatial decomposition on the residual image, the image is down-sampled to lower resolutions. Therefore, the spatial scalability is inherent in the interframe wavelet video coder. However, the spatial scalability is not fine-tuned scalable. For an original frame size of m-by-n, the spatial scalability of the image is restricted to the size of $m/2^p$-by-$n/2^p$ where p is an integer.

The truncation process keeps the information of subbands that are lower than or equal to the required spatial resolution and truncates the other subbands.

Upon decoding, the motion vectors of the motion information are scaled by the factor of $p$, regarding the rescaled size. The residual image data are then motion compensated temporal synthesized with the scaled set of motion vectors to reconstruct

the original sequence.

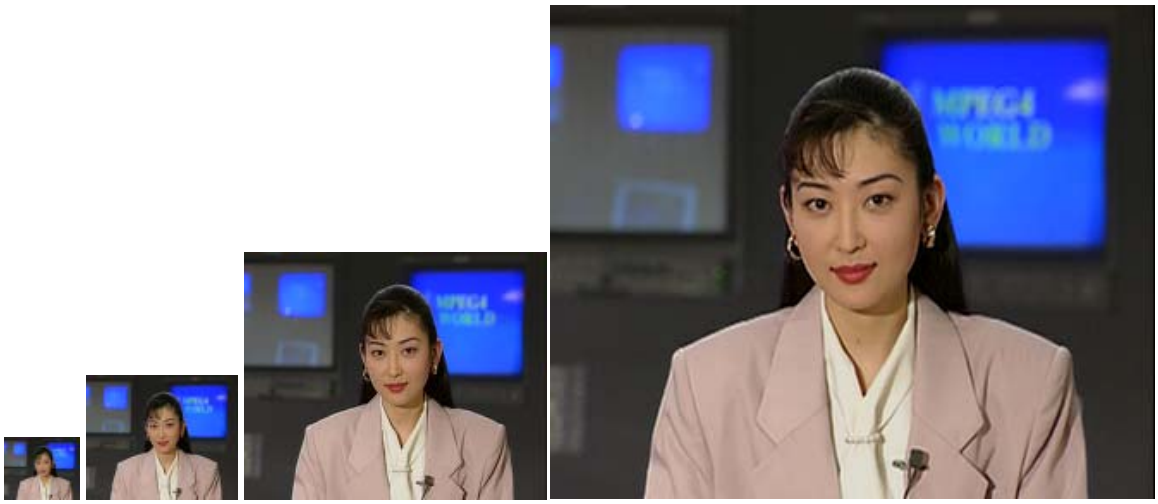
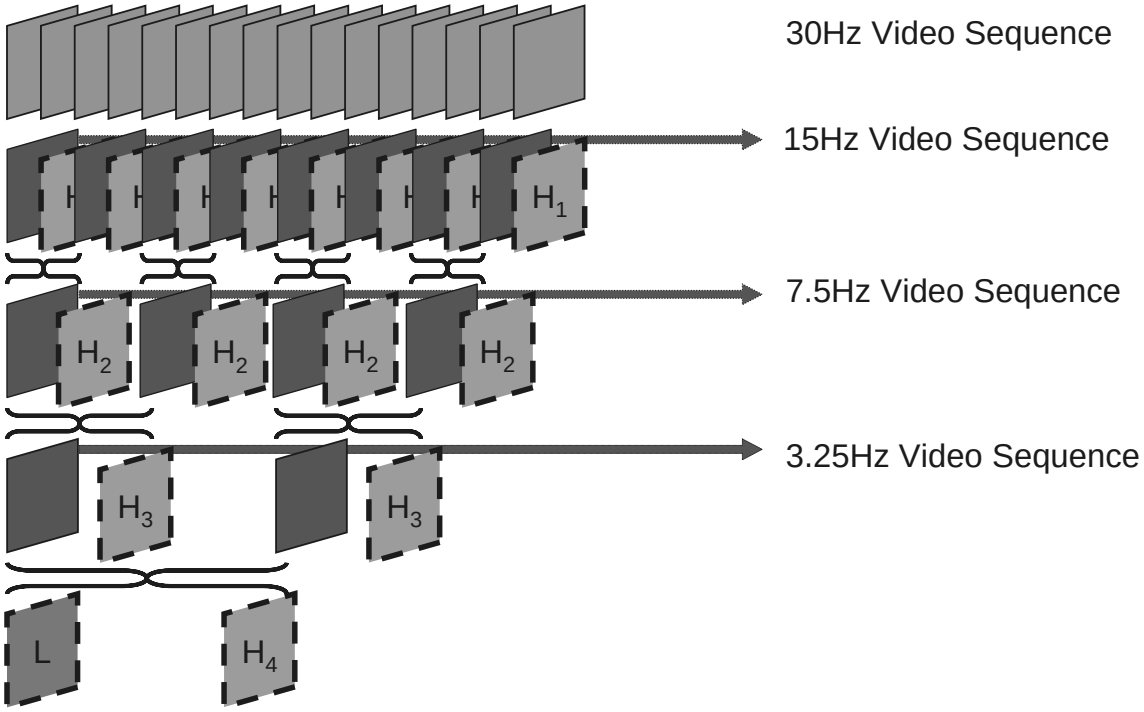


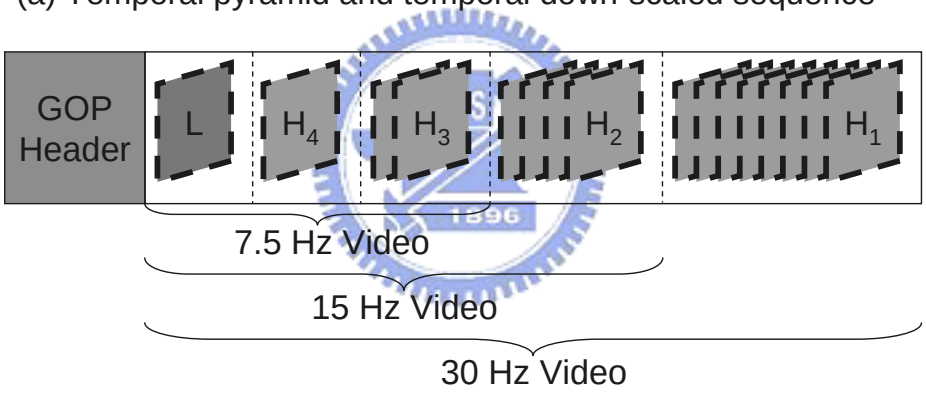
Figure 2-19 Spatial scalability.

## 2.4.3 Temporal Scalability

The interframe wavelet transform will create a temporal pyramid after MCTF. In order to reduce the amount of transform data, we can discard the temporal high-pass frame, as that shown in Figure 2-20(a). To achieve temporal scalability, the truncation process keeps the subset of images that are needed to generate the required level of temporal pyramid, as that shown in Figure 2-20(b).




(a) Temporal pyramid and temporal down-scaled sequence




(b) The GOP of the temporal scaled sequence

Figure 2-20 Temporal scalability.

If motion estimated motion trajectory is not perfectly matched to the original video sequence, the temporal filtering process might generate some motion artifacts [14].

# Chapter 3
# 3D Subband Video Coding Using Barbell Lifting

In 68th MPEG meeting (March, 2004, Munich), MSRA proposed its MCTF structure and 3D ESCOT entropy coder. The 3D ESCOT entropy coder performs almost as well as the 3D EBCOT that JPEG2000 used. Figure 3-1 shows the block diagram of this coding structure [15]. This proposed video coding algorithm has two different concepts. They are Barbell lifting and 3D ESCOT entropy coding. The motion estimation scheme of this video coding algorithm is not HVSBM but a motion estimation scheme used in H.264. We will describe them in the following subsections.

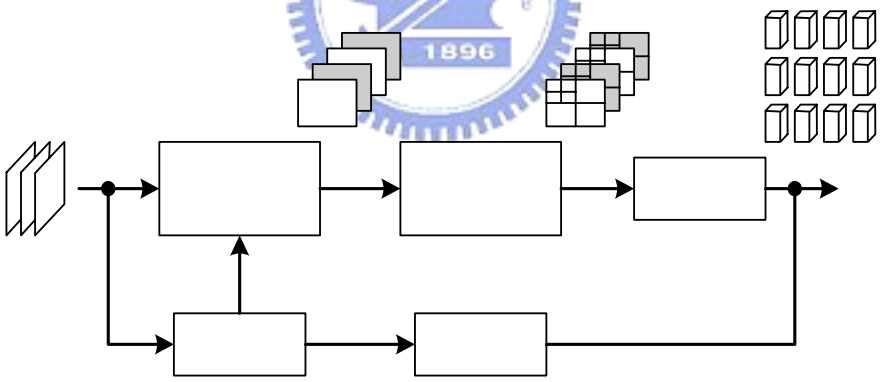


Figure 3-1 The block diagram of the 3D subband video coding using Barbell lifting [15].

## 3.1 Barbell Lifting

MSRA proposes this Barbell lifting algorithm for doing temporal decomposition [15]. Barbell lifting uses a set of pixels instead of a pixel as the input, as that shown in Figure 3-2. The Barbell lifting can provide perfect reconstruction, sub-sample decomposition but still with critically sampled transformed coefficients.

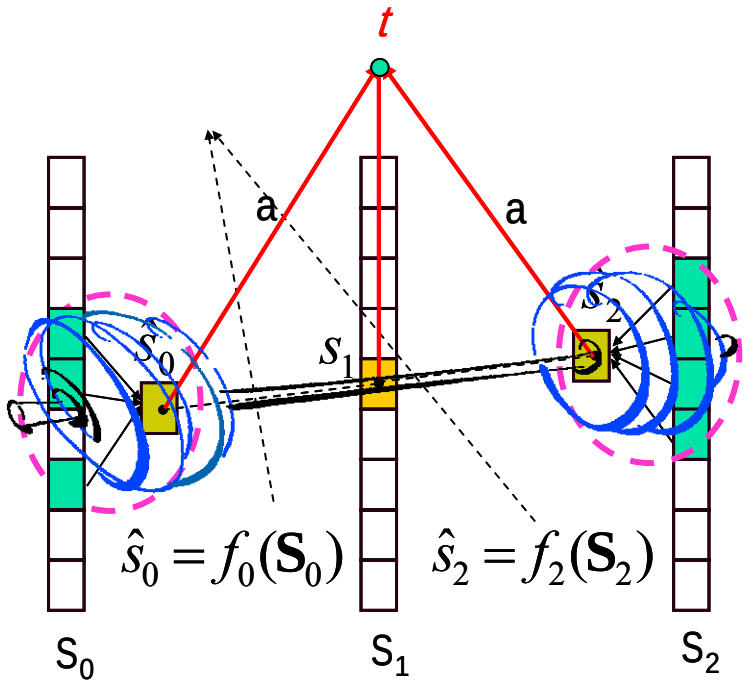



Figure 3-2 The Barbell lifting [15].

Assume that $S_0$, $S_1$, and $S_2$ are three consecutive frames in a video sequence. Functions $f_0()$ and $f_2()$ are called as Barbell functions and they can be any linear or non-linear functions that take any pixels in the same frames as variables. The Barbell functions can also vary from pixel to pixel. Therefore the basic Barbell lifting step is formulated as:

$$t = a \times \hat{s}_0 + s_1 + a \times \hat{s}_2, \tag{6}$$

where $a$ is a filter parameter.

The Barbell lifting includes two stages. They are prediction stage and update stage. The prediction stage is applied to the video sequence first. It takes the original input frames to generate the high-pass frames, as shown in Figure 3-3.

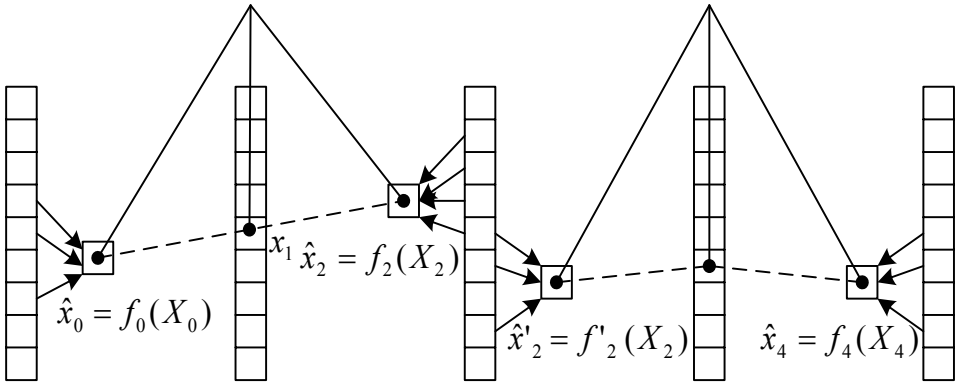
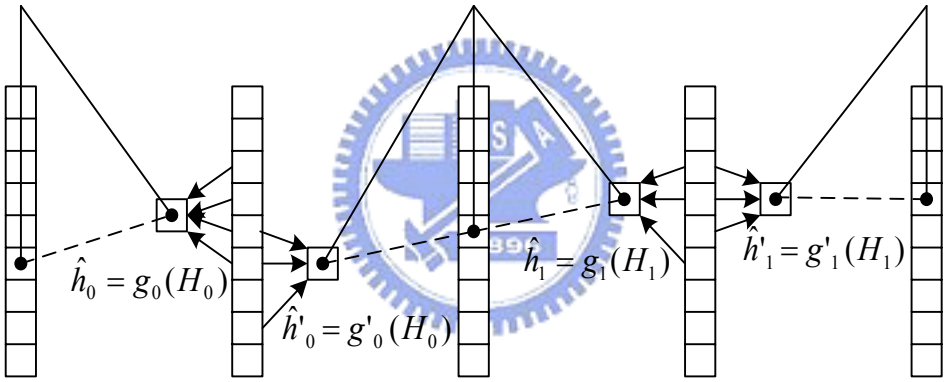



Figure 3-3 The prediction stage of the Barbell lifting.

Then the update stage uses the available high-pass frames and the even frames to generate the low-pass frames, as shown in Figure 3-4.




-a

Figure 3-4 The update stage of the Barbell lifting.

## 3.1.1 The Prediction Stage

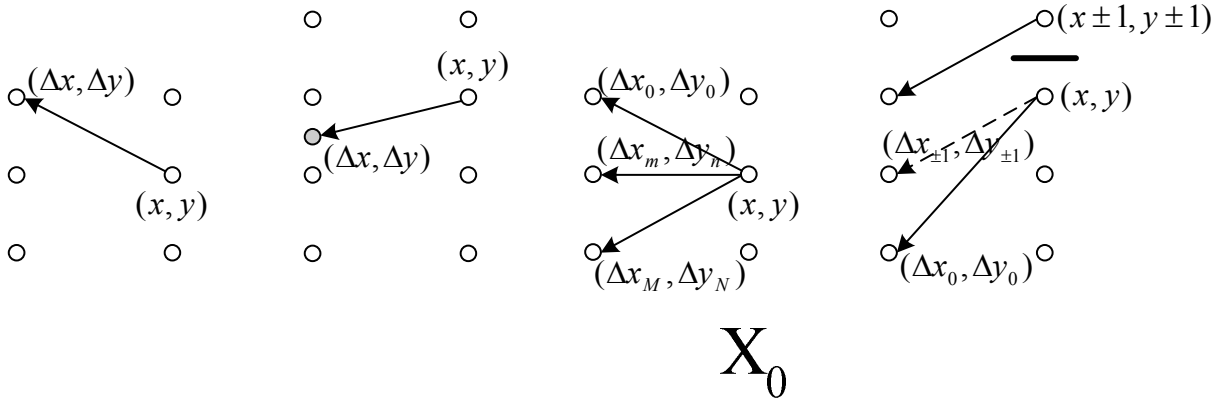



Figure 3-5 The Barbell functions used in the prediction stage.

Figure 3-5 shows some examples of Barbell functions used in the prediction stage and Figure 3-5(a) is the integer motion alignment case and the Barbell function of this case is:

$$f = F_i(x + \Delta x, y + \Delta y), \tag{7}$$

where $(\Delta x, \Delta y)$ is the motion vector of current pixel (x, y) and $F_i$ is the previous frame.

Figure 3-5(b) is the fractional-pixel motion alignment case and the Barbell function of this case is:

$$f = \sum_m \sum_n \alpha(m,n) F_i(x + \lfloor \Delta x \rfloor + m, y + \lfloor \Delta y \rfloor + n), \tag{8}$$

where $\lfloor \ \rfloor$ denotes the integer part of $\Delta x$ and $\Delta y$. $\alpha(m,n)$ is the factor of the interpolation filter.

Figure 3-5(c) is the multiple-to-one mapping case and the Barbell function of this case is:

$$f = \sum_m \sum_n \alpha(m,n) F_i(x + \Delta x_m, y + \Delta y_n), \tag{9}$$

where $\alpha(m,n)$ is the weighting factor for each connected pixel.

Figure 3-5(d) shows a special case that the current pixel (*x, y*) can use its motion vector $(\Delta x, \Delta y)$ and the motion vectors of neighboring pixels to get multiple predictions from the previous frame and generate a new prediction. The Barbell function of this case is:

$$f = \sum_{m=0,\pm1} \sum_{n=0,\pm1} \alpha(m,n) F_i(x + \Delta x_m, y + \Delta y_n), \tag{10}$$

where $\alpha(m,n)$ is the weighting factor.

## 3.1.2 The Update Stage

The prediction and update stages may has mismatch when pixels in different frames are aligned with motion vectors at fractional-pixel precision or without one-to-one

mapping. Generally speaking, the update and prediction stages use the same motion vector for saving overhead bits to code motion vectors, i.e., the motion vector of the update stage is the inverse one of the prediction stage. Figure 3-6 shows the mismatch problem.

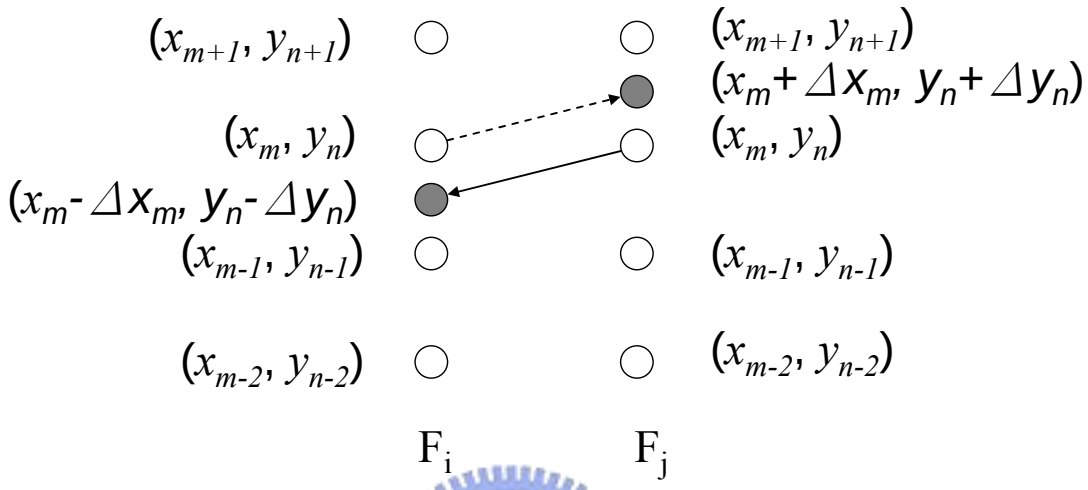



Figure 3-6 The mismatch problem of motion in the prediction and update stages.

As shown in Figure 3-6, the mismatch problem is that the prediction has the path from $F_i(x_m, y_n)$ to $F_j(x_m + \Delta x_m, y_n + \Delta y_n)$ but the update has the path from $F_j(x_m, y_n)$ to $F_i(x_m - \Delta x_m, y_n - \Delta y_n)$.

Barbell lifting can solve this mismatch problem. In the update stage, the obtained high-pass coefficients are likely distributed to those pixels that are used to calculate the high-pass coefficient in the prediction stage. Assuming that equation (9) is the Barbell function used in the prediction stage now. We can calculate the high-pass coefficients by combining equations (6) and (9). Then we calculate the high-pass coefficients by:

$$h_j(x,y) = F_j(x,y) + \sum_i \sum_m \sum_n a_i \alpha_{i,j}(x,y,m,n) F_i(x + \Delta x_m, y + \Delta y_n), \qquad (11)$$

where $\alpha_{i,j}(x,y,m,n)$ is the Barbell parameter specified by the coordination *x, y, m, n*. Then we can calculate low-pass coefficients in the same way by:

$$l_i(x,y) = F_i(x,y) + \sum_j \sum_m \sum_n b_j \alpha_{i,j}(x,y,m,n) h_j(x + \Delta x_m, y + \Delta y_n). \quad (12)$$

It means that the high-pass coefficient will be added back exactly to the pixels that are predicted.

For the above example, the predicted weight from $F_i(x_{m-1}, y_{n-1})$ to $F_j(x_m, y_n)$ is non-zero. So in the proposed technique, the update weight from $F_j(x_m, y_n)$ to $F_i(x_{m-1}, y_{n-1})$, which equals to the predict weight, is also not zero.

## 3.2 Spatial Decomposition

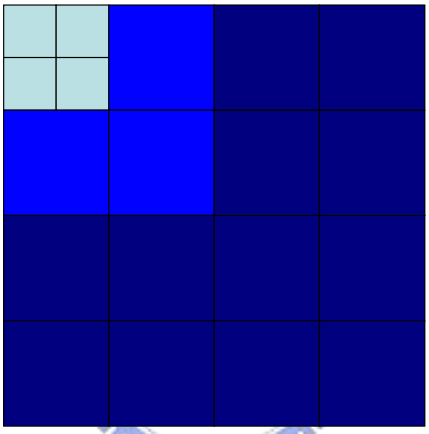


Figure 3-7 The frame after 3 level spatial decomposition.

After temporal decomposition, the spatial decomposition is applied to each created residual frame. The filter used here is the Daubechies 9/7 filter and the analysis filter coefficients are shown in Table 3-1 [35]. The coefficients of the Daubechies 9/7 synthesis filter are shown in Table 5-1 [35].

| index | Analysis low pass filter | Analysis high pass filter |
|---|---|---|
| 0 | 0.6029490182363579 | 1.115087052456994 |
| ±1 | 0.2668641184428723 | -0.5912717631142470 |
| ±2 | -0.07822326652898785 | -0.05754352622849957 |
| ±3 | -0.01686411844287495 | 0.09127176311424948 |
| ±4 | 0.02674875741080976 | |

Table 3-1 The coefficients of the Daubechies 9/7 analysis filters.

The spatial decomposition can also be done on the LH, HL, and HH subbands of the first level decomposition. Thus we can get the important information in those subbands and code them.

## 3.3 Multi-Layer Motion Estimation and Coding

The video coding algorithm proposed by MSRA dose not use HVSBM in motion estimation. It uses a motion estimation method adopted in H.264 but makes some changes to achieve motion information scalability.

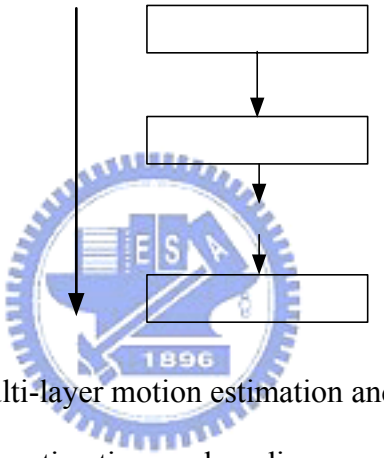


Figure 3-8 Multi-layer motion estimation and coding.

It uses multi-layer motion estimation and coding as shown in Figure 3-8. It generates an embedded bitstream for motion, which consists of one base layer and a few enhancement layers. A coarse motion field can be reconstructed from the base layer and can be successively refined by subsequent enhancement layers. The motion vectors of the base layer are large and coarse and may be used for low bit rates. The motion vectors of enhancement layer are small with details and often used for high bit rates.

## 3.4 3D ESCOT

After temporal and spatial decomposition, the generated coefficients will be coded with 3D Embedded Subband Coding with Optimal Truncation (3D ESCOT) [16]. The

3D ESCOT is in principle very similar to EBCOT used in JPEG2000 [9], which deals with 2D image coding. We can call 3D ESCOT as 3D EBCOT because it is an extension of EBCOT used to do 3D dimensional signal coding. 3D ESCOT can offers high compression efficiency and other functionalities, such as error resilience and random access.

3D ESCOT takes advantages of the orientation-invariant property of wavelet subbands to reduce the number of context and codes each subband independently so each subband can be decoded independently. Because of this feature, 3D ESCOT can achieve flexible spatial/temporal scalability and R-D optimization can be done within subbands to improve compression efficiency.

Each subband is divided into 3D coding blocks and these blocks are coded independently.

For each coefficient x[i, j, k] at position [i, j, k], we assign it a binary-valued state variable σ[i, j, k], which indicates the significance of this coefficient. χ[i, j, k] is defined as the sign of the x[i, j, k]. It is 0 when the sample is positive and 1 when the sample is negative. σ[i, j, k] is initialized to 0 and toggled to 1 when the x[i, j, k]'s first non-zero bit-plane value is encoded. There are three coding operations and when they will be used depends on σ[i, j, k]. Zero coding (ZC) and sign coding (SC) will be used to code x[i, j, k] if σ[i, j, k] = 0 and magnitude refinement (MR) will be used if σ[i, j, k] = 1. We will introduce these three coding operations as follows.

### 3.4.1 Zero Coding

If a coefficient x[i, j, k] is not yet significant in the previous but-planes, i.e., σ[i, j, k] = 0, ZC is used to code the new information about whether it becomes significant or not in the current bit-planes. ZC uses significant information about x[i, j, k]'s immediate neighbors as the context to code the its own significant information. There

are four types of neighbors as shown in Figure 3-9.

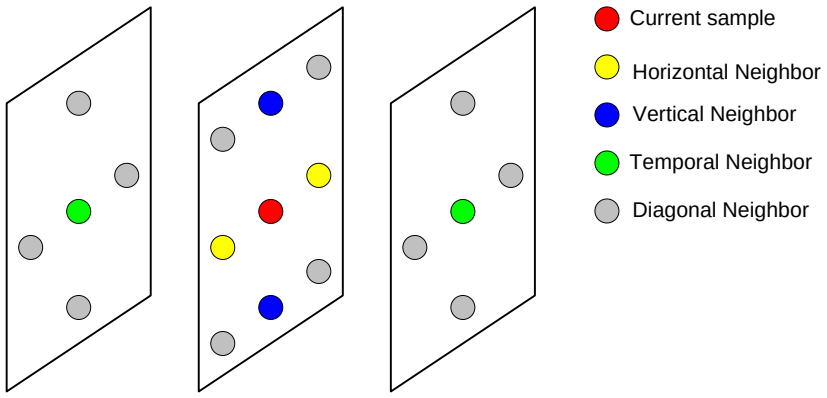



Figure 3-9 Four types of coding neighbors for zero coding.

1. Immediate horizontal neighbors. The number of these neighbors is 2 and the number of significant ones is denoted by h, $0 \leq h \leq 2$.
2. Immediate vertical neighbors. The number of these neighbors is 2 and the number of significant ones is denoted by v, $0 \leq v \leq 2$.
3. Immediate temporal neighbors. The number of these neighbors is 2 and the number of significant ones is denoted by a, $0 \leq a \leq 2$.
4. Immediate temporal neighbors. The number of these neighbors is 12 and the number of significant ones is denoted by d, $0 \leq d \leq 12$.

Table 3-2 shows the context assignment map of ZC. If the conditions of two or more rows are satisfied in the same time, the low-numbered context is selected.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LLL and LLH sub-band | h | 2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | v | x | ≥1 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 |
| | a | x | x | ≥1 | 0 | 0 | 0 | ≥1 | 0 | 0 | 0 | 0 |
| | d | x | x | x | x | x | x | x | 3 | 2 | 1 | 0 |
| | context | 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| LHH sub-band | h | 2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| | v+a | x | ≥3 | ≥1 | ≥1 | 0 | 0 | ≥3 | ≥1 | ≥1 | 0 | 0 |
| | d | x | x | ≥4 | x | ≥4 | x | x | ≥4 | x | ≥4 | x |
| | context | 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

| HHH sub-band | d | ≥6 | ≥4 | ≥4 | ≥2 | ≥2 | ≥2 | ≥0 | ≥0 | ≥0 | ≥0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | h+v+a | x | ≥3 | x | ≥4 | ≥2 | x | ≥4 | ≥2 | 1 | 0 |
| | context | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Table 3-2 Context assignment map for ZC.

## 3.4.2 Sign Coding

SC is called to code χ[i, j, k], which is the sign of coefficient x[i, j, k], if x[i, j, k] becomes significant in the current bit-plane. SC also utilizes high-order context-based arithmetic coding to compress the sign symbols. The context models of arithmetic coding are based on three quantities $h_s$, $v_s$ and $t_s$. They are defined as follows:

$$h_s=\min\{1, \max\{-1, \sigma[i-1,j,k] \times (1-2\chi[i-1,j,k])+ \sigma[i+1,j,k]\times(1-2\chi[i+1,j,k])\}\}, \quad (13)$$

$$v_s=\min\{1, \max\{-1, \sigma[i,j-1,k] \times (1-2\chi[i,j-1,k])+ \sigma[i,j+1,k] \times (1-2\chi[i,j+1,k])\}\}, \quad (14)$$

$$t_s=\min\{1, \max\{-1, \sigma[i,j,k-1] \times (1-2\chi[i,j,k-1])+ \sigma[i,j,k+1] \times (1-2\chi[i,j,k+1])\}\}. \quad (15)$$

Table 3-3 shows the context assignment map and sign prediction map of SC. $\hat{\chi}$ is the sign symbol prediction under the given context and the symbol sent to the arithmetic coder is $\hat{\chi} \oplus \chi$

| $h_s$=-1 | $v_s$ | -1 | -1 | -1 | 0 | 0 | 0 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| | $t_s$ | -1 | 0 | 1 | -1 | 0 | 1 | -1 | 0 | 1 |
| | $\hat{\chi}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | context | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| $h_s$ =0 | $v_s$ | -1 | -1 | -1 | 0 | 0 | 0 | 1 | 1 | 1 |
| | $t_s$ | -1 | 0 | 1 | -1 | 0 | 1 | -1 | 0 | 1 |
| | $\hat{\chi}$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| | context | 9 | 10 | 11 | 12 | 13 | 12 | 11 | 10 | 9 |

| $h_s = 1$ | $v_s$ | -1 | -1 | -1 | 0 | 0 | 0 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| | $t_s$ | -1 | 0 | 1 | -1 | 0 | 1 | -1 | 0 | 1 |
| | $\hat{\chi}$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | context | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Table 3-3 Context assignment and sign prediction map for SC.

### 3.4.3 Magnitude Refinement

MR is called to code new information about x[i, j, k] if σ[i, j, k] was switched to 1 in the previous bit-plane, i.e., it becomes significant. It uses three contexts for arithmetic coding.

1. The context of x[i, j, k] is 0 if MR not yet used for x[i, j, k].
2. The context of x[i, j, k] is 1 if MR has been used for x[i, j, k] and x[i, j, k] has at least one significant neighbor by now.
3. Otherwise, the context is 2.

### 3.4.4 Fractional Bit-Plane Coding

The practical coding gain of 3D ESCOT is higher than 3D SPIHT because SC and MR have high-order context modeling and the use of fractional bit-plane coding [16]. The fractional bit-plane coding can provides a practical means of scanning the wavelet coefficients within each bit-plane for rate-distortion (R-D) optimization at different rates. There are three different fractional bit-plane passes and the scanning order in each of them is along the i-direction firstly, then the j-direction and the k-direction lastly.

#### 3.4.4.1 Significance Propagation Pass

If the coefficients which are not yet significant but have “preferred neighborhood” are processed by this pass. A coefficient has a “preferred neighborhood” if and only if

the coefficient has at least one significant immediate diagonal neighbor for HHH subband or horizontal, vertical, temporal neighbor for the other types of subband. For these coefficients, we apply the ZC to code their significance information in the current bit-plane of this coefficient. If the coefficient becomes significant in the current bit-plane, then SC is used to code the sign.

### 3.4.4.2 Magnitude Refinement Pass

If the coefficient became significant in the previous bit-plane, it will be coded in this pass. The binary bits corresponding to these coefficients in the current bit-plane are coded by MR.

### 3.4.4.3 Normalization Pass

It is used to code the coefficients if it was not coded in the previous two passes. Because these coefficients are not yet significant, they are only processed by ZC and SC.

## 3.5 Bitstream Truncation and Scalability

After 3D ESCOT on each subband, an embedded bitstream is generated for each subband. In order to satisfy the requested bit rate, bitstreams corresponding to different subbands will be truncated and multiplexed together to construct final bitstream then transmitted to the receiver. The rate control problem is how to truncate and multiplex bitstreams to create the final bitstream that achieves the best R-D optimization.

The basic problem of rate control is that given a target bit rate $R_0$, how to construct a bitstream that satisfies the bit rate constraint and minimizes the overall distortion. Shoman and Gersho proposed a Lagrange's theorem that can solve this problem [17].

Taubman extends this algorithm to the rate control of EBCOT [9].

EBCOT partitions the subbands representing the image into a collection of relatively small code-blocks, $B_i$, whose embedded bitstreams may be truncated to the rate $R_i^n$. The contribution from $B_i$ to the distortion in the reconstructed image is denoted $D_i^n$, for each truncation point $n$. Assuming that the distortion of each code-block is independent and additive. Thus the overall reconstructed image distortion $D$ can be represented by:

$$D = \sum_i D_i^{n_i} , \tag{16}$$

where $n_i$ denotes the truncation point selected for code-block $B_i$. $D_i^n$ is calculated by:

$$D_i^n = w_{b_i}^2 \sum_{k \in B_i} (s_i[k] - s_i^n[k])^2 , \tag{17}$$

where $s_i[k]$ is the 2D sequence of subband coefficients in code-block $B_i$. $s_i^n[k]$ is the quantized representation of these coefficients associated with truncation point $n$, and $w_{b_i}$ is the L2-norm of the wavelet basis functions for the subband, $b_i$, to which code-block $B_i$ belongs.

R-D optimization algorithm should select truncation points $n_i$ for each code-block $B_i$ such that the sum of $R_i^{n_i}$ or $D_i^{n_i}$ meets the constraint imposed by $R_{max}$ or $D_{max}$ and also the sum of $D_i^{n_i}$ or $R_i^{n_i}$ is the minimum value. They are described as follows:

$$\sum_i D_i^{n_i} = D = D_{\min} \text{ , given } \sum_i R_i^{n_i} = R \le R_{\max} , \tag{18}$$

or

$$\sum_i R_i^{n_i} = R = R_{\min} \text{ , given } \sum_i D_i^{n_i} = D \le D_{\max} . \tag{19}$$

Recently, several R-D optimization algorithms have been proposed to solve this

problem [18]. It is noticeable that all these algorithms are applicable to convex curves. Convex curves are the curves that the slopes are strictly decreasing. Some R-D optimization algorithms are based on Lagrange's theorem, such as the Lagrange multiplier used in EBCOT [9]. Lagrange's theorem states that the sum of continuous functions with boundary condition is optimized at the points with equal slopes as shown below:

$$(D(\lambda)+\lambda R(\lambda)) = \sum_i (D_i^{n_i^\lambda} + \lambda R_i^{n_i^\lambda}) . \tag{20}$$

Any set of truncation points, $\{n_i^\lambda\}$, which minimizes $(D(\lambda)+\lambda R(\lambda))$ for some $\lambda$ is optimal in the sense that the distortion cannot be reduced without increasing the overall rate or vice-versa. If we can find a value of $\lambda$ such that the truncation points minimize $(D(\lambda)+\lambda R(\lambda))$ yields $R(\lambda) = R_{\max}$, then this set of truncation points must be an optimal solution to the R-D algorithm based on Lagrange's theorem.

Because the number of truncation points in a code-block is finite, we can not find the value of $\lambda$ such that $R(\lambda)$ exactly equals to $R_{\max}$. However, since the code-block in EBCOT is very small such that the total number of truncation points is very large, we can find the smallest value of $\lambda$ such that $R(\lambda) \le R_{\max}$.

In order to find the optimal truncation point sets $n_i^\lambda$ for any given $\lambda$, we need to know the rate-distortion (R-D) pair of each truncation points. $\lambda$ can be viewed as the R-D slope of the optimal truncation point sets. We can find the R-D slope of each truncation point by calculating the bitstrean length and distortion at that point. Thus we can construct an operational R-D curve for each code-block.

1) Assume $n$ is the number of the truncation points, and $0 \leqq j \leqq n$.
2) For $j$ = 0, 1, 2, …, $n$, 0 is the beginning of the code-block, not a truncation point.

   The R-D slope of each truncation point $j$ is $\frac{\Delta D_i^j}{\Delta R_i^j} = \frac{D_i^{j-1} - D_i^j}{R_i^j - R_i^{j-1}}$, where $R_i^j$ is the

accumulative bit length of truncation point $j$ in code block $i$ and $D_i^j$ is the accumulative distortion of truncation point $j$ in code block $i$.

Generally speaking, $R_i^j \geq R_i^{j-1} \geq R_i^{j-2} \geq ... \geq R_i^0 = 0$ and $D_i^j \leq D_i^{j-1} \leq D_i^{j-2} \leq ... \leq D_i^0 =$ the distortion when the coefficients of the code-block are all 0. We just need to package the truncation points with the R-D slope bigger than or equal to $\lambda$, then we can achieve the optimal R-D.

In 3D ESCOT, the end of each fractional bit-plane is a candidate truncation point. The R-D slope of each truncation points can be obtained by calculating the bitstrean length and distortion [16]. Then we can construct an operational R-D curve for each subband and find its convex hull. All valid truncation points must lie on this convex hull such that the R-D optimality at each truncation point can be guaranteed. If the truncation point does not have a strictly decreasing R-D slope (i.e., it has larger distortion than the previous truncation point), it will be discarded. In order to find the best threshold value $\lambda$, we first set an arbitrary value of $\lambda$. If the R-D slope of this truncation point is bigger than or equal to $\lambda$, this truncation point will be packaged. After we process all of the truncation points, we obtain the final bitstream. If the bit rate of this bit-stream is larger than that of requested, the value of $\lambda$ will be set larger to find the final bitstream again. Otherwise, the value of $\lambda$ will be set smaller. We use this method recursively to find the final bitstream that has bit rate smaller than or equal to the requested bit rate.

# Chapter 4
# Human Visual System

## 4.1 Human Vision

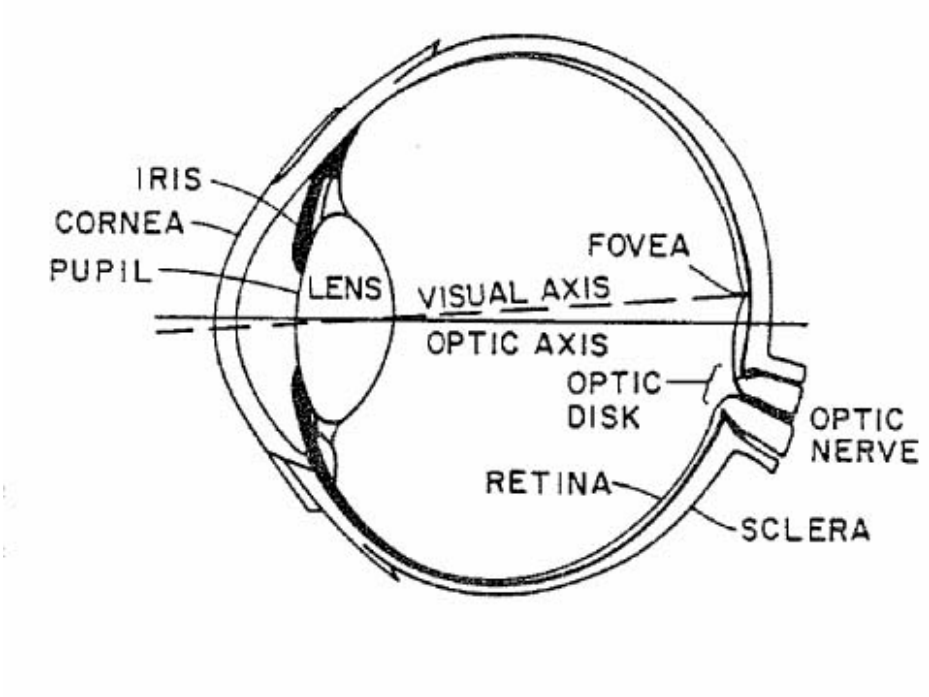



Figure 4-1 Cross-section of human eye [19].

Figure 4-1 shows the cross-section of a human eye [19]. Through the optics of the eye, the visual input is projected onto the retina, the neural tissue at the back of eye composed of the photoreceptor mosaic [20]. The photoreceptors sample the image and convert the input image to the signals that can be interpreted by the visual cortex of the brain. Photoreceptors have Rhodopsin which is very sensitivity to light. When Rhodopsin receives the energy of light, it will decompose into Vitamins A, Protein, and impulse signal. The impulse signal will be processed by the Bipolar cell and Ganglion cell then passed through optical nerves into the brain as shown in Figure 4-2 [21]. The Vitamins A, Protein, and Nutrition will be combined together and converted

to Rhodopsin by the effect of Enzyme. Then the Rhodopsin can be used again.

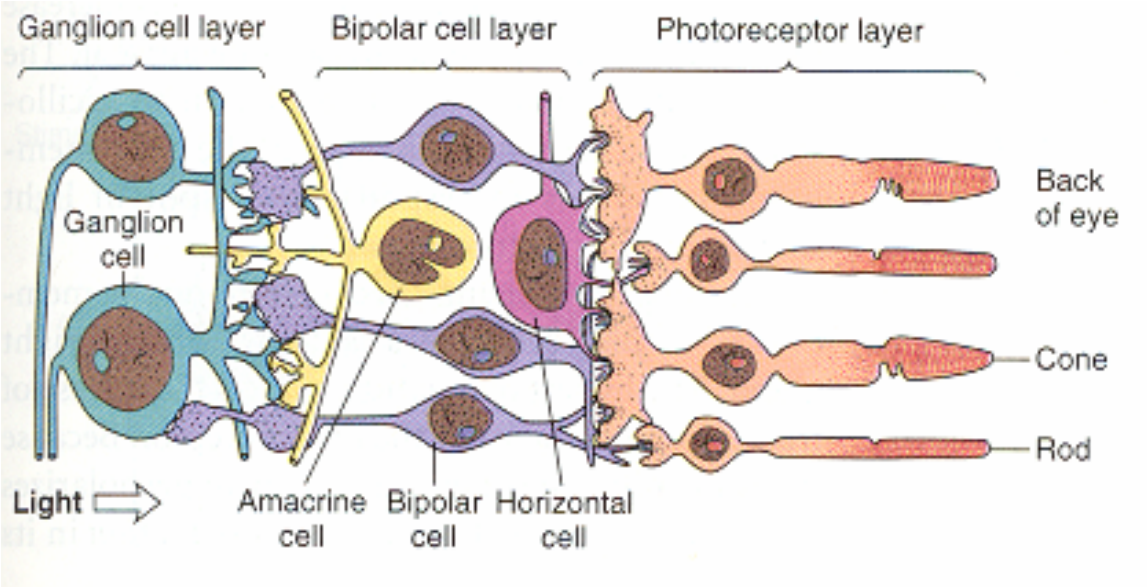



Figure 4-2 The process of the visual input signal [21].

There two types of photoreceptors, rods and cones. Rods are relatively long and thin. They are used to view at lower several orders of magnitude of illumination, i.e., under scotopic conditions. Cones are relatively shorter and thicker and they are less sensitive than rods. They are used to view at the higher 5 to 6 orders of magnitude of illumination, i.e., under photopic conditions. The cones are concentrated in the fovea, the region of highest visual acuity, which covers approximately two degrees of visual angle on the retina. The cones are also responsible for color vision.

There three types of cones. They are L-cones, M-cones, and S-cones. L-cones are also called Red cones and they are sensitive to long wavelengths. M-cones are also called Green cones and they are sensitive to medium wavelengths. S-cones are also called Blue cones and they are sensitive to short wavelengths. Figure 4-3 shows the relative sensitivity of each photoreceptor [21].

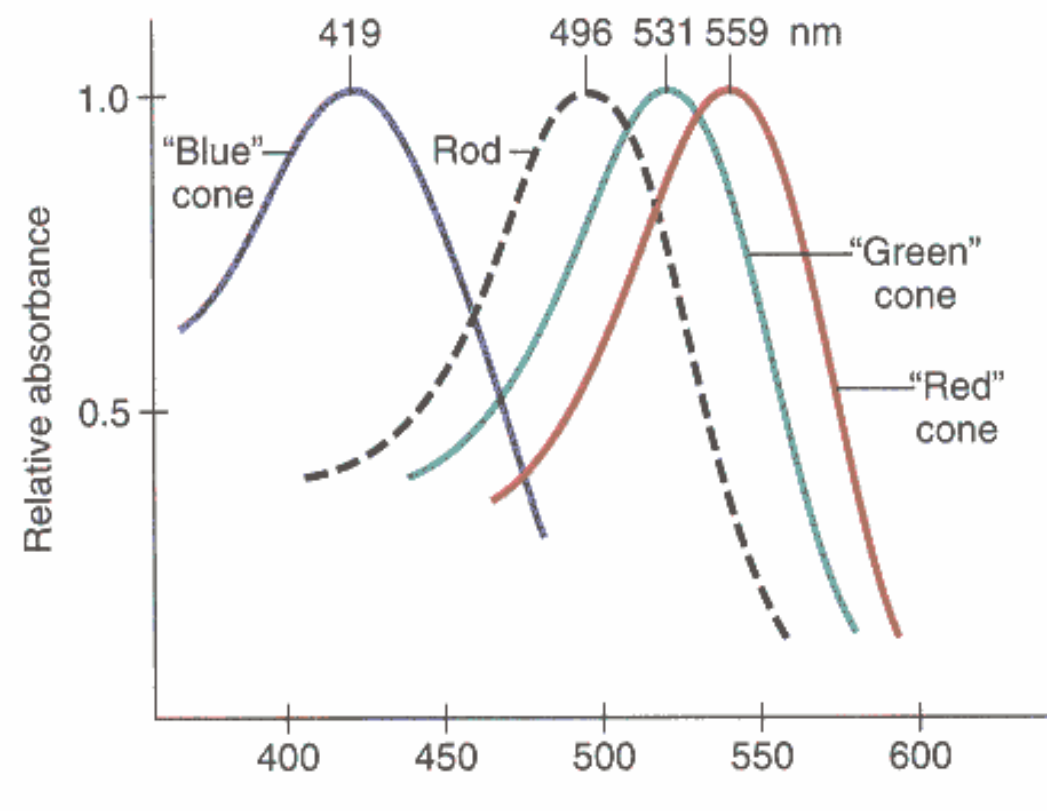



Figure 4-3 Relative sensitivity of each photoreceptor [21].

## 4.2 Color Representation

Colors do not exist in natural world. To human perception, colors are related to the wavelength of light. As describes above, the retina of human eye contains 3 different color receptors: red, green, and blue. The different cones have different sensitivity curve to light of different frequency. Thus, the combination of different sensitivity curve to light can produce different color recognition. Due to this structure of human eye, any color appeared to human eye can be specified by a weighted combination of three so-called primary colors RGB. For the purpose of standardization, the CIE (Commission Internationale de L'eclairage─ International Commission on Illumination) chooses the following specific wavelength values to the three primary colors: blue (**B**) = 435.8nm, green (**G**) = 546.1nm, and red (**R**) = 700.0nm.

Trichromatic theory says that any color *S* can be represented as a combination of

these 3 primaries **R, G,** and **B**.

$$S = R_s \cdot \mathbf{R} + G_s \cdot \mathbf{G} + B_s \cdot \mathbf{B}. \tag{21}$$

Any 3 independent colors can be selected as primaries as long as one is not a mix of the other two. Different sets of primaries are related by linear transformations.

There several color models, such as CIE RGB, CIE XYZ, CIE YUV, and CIE L*a*b*. We introduce CIE RGB and CIE XYZ here.

1. CIE RGB:

1) R, G, B = three spectral primary source.

2) Reference white: R = G = B = 1.

3) There exist negative tristimulus values.

4) The color is fully dependent on the wavelength. The three fixed RGB components acting alone cannot generate all spectrum colors (pure colors). This is an unresolved defect for color representation.

2. CIE XYZ

1) All color matching functions are positive.

2) Y = luminance

3) Reference white: X = Y = Z = 1.

4) This model is modified from RGB model such that all spectral tristimulus values are positive.

Generally Speaking, Each color space can transform to another space. Equation (22) is the transformation from CIE RGB to CIE XYZ and equation (23) is CIE XYZ to CIE RGB.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 2.365 & -0.515 & 0.005 \\ -0.897 & 1.426 & -0.014 \\ -0.468 & 0.089 & 1.009 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{22}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 0.490 & 0.177 & 0.000 \\ 0.310 & 0.813 & 0.010 \\ 0.200 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \quad (23)$$

## 4.3 Contrast Sensitivity

Human perception is more sensitive to the contrast of the luminance than the absolute value of the luminance. But due to the complexity of natural image, a common definition of contrast suitable for all conditions does not exist. Generally speaking, there are three types of contrast definitions widely used.

In the case of a periodic pattern of symmetrical deviations ranging from $L_{min}$ to $L_{max}$, Michelson contrast is generally used:

$$C_M = \frac{L_{\max} - L_{\min}}{L_{\max} + L_{\min}}. \quad (24)$$

When the pattern consists of a single increment or decrement $\Delta L$ to an otherwise uniform background luminance $L$, Weber contrast is often used:

$$C_W = \frac{\Delta L}{L}. \quad (25)$$

These two definitions of contrast are not appropriate for measuring the contrast of complex images. If there are some very bright or very dark points in the image, these points will determine the contrast of the whole image. Furthermore, human contrast perception varies with the local average luminance. Peli proposed a local band limited contrast measure to solve these problems [22]:

$$Ci(x,y) = \frac{BP_i(x,y)}{LP_i(x,y)}, \quad (26)$$

where $BP_i(x,y)$ is the bandpass image of band $i$ at location$(x, y)$, and $LP_i(x,y)$ contains the energy below band $i$ at location $(x, y)$, i.e., the total response at this location of all the bands below the band $i$. Modifications of this contrast definition have been used in a number of vision models and are in good agreement with

psychophysical experiments on Gabor patches [38].

We can describe contrast sensitivity as the function of spatial frequency. This function is called contrast sensitivity function (CSF). Contrast sensitivity is defined as the inverse of contrast threshold. The contrast threshold is the minimum contrast necessary for an observer to detect the target.

Mannos and Sakrison first applied the HVS to image coding. They model the HVS as a nonlinear point transform followed by the modulation transform function (MTF) of the form [23]:

$$H(f) = 2.6(0.192 + 0.114f)\exp(-(0.114f)^{1.1}) . \tag{27}$$

Nill proposed a new type of MTF that can be used for DCT [24]:

$$H(f) = (0.2 + 0.45f)\exp(-0.18f) . \tag{28}$$

Ngan et al proposed another new MTF [25]:

$$H(f) = (0.31 + 0.69f)\exp(-0.29f) . \tag{29}$$

Except for the dependence on spatial frequency, the contrast sensitivity also depends on temporal frequency. Thus we can describe contrast sensitivity as the function of spatial frequency and temporal frequency. Kelly proposed a contrast sensitivity function (CSF) and it is generally used [26]:

$$CSF(f_s, f_t) = 4\pi^2 f_s f_t \exp\left(\frac{-4\pi(f_t + 2f_s)}{45.9}\right) \times \left(6.1 + 7.3\left|\log\left(\frac{f_t}{3f_s}\right)\right|^3\right) . \tag{30}$$

From this CSF, we can see that human has lower sensitivity at low and high spatial (temporal) frequency but higher sensitivity at medium spatial (temporal) frequency.

## 4.4 Masking Effect

If a stimulus can be visible by itself but can not be detected due the presence of another stimulus, this effect is called masking effect. On the other hand, the opposite

effect, facilitation, occurs when a stimulus can not be visible itself can be detected due to the presence of another stimulus. Masking effect explains why similar coding artifacts are disturbing in certain regions of an image while they are hardly noticeable elsewhere. There two types of masking effect, spatial masking and temporal masking.

Spatial masking is due to the non-uniformity of the background luminance. Because of this masking effect, the noise is more visible in the flat or texture-less areas and less visible in region with edges and textures. So the coding errors may be less visible around sharp edges.

Temporal masking is due to the temporal discontinuity in intensity, like scene change. The error visibility threshold is increased with the increasing interframe luminance difference. Sometimes, if moving objects are not tracked by eyes, the loss of perceived spatial resolution is substantial.

## 4.5 Just-Noticeable Distortion

The definition of just-noticeable distortion (JND) is the visibility threshold of distortion and the reconstruction errors below this threshold are imperceptible [27]. Sometimes we use the inverse of the sensitivity as the threshold. Human eyes are more sensitive to luminance contrast than to absolute luminance value. The detecting ability of human eyes to the difference between objects and background depends on average value of background luminance. Weber's law said that the ration of just noticeable luminance difference to stimulus' luminance is almost constant if the luminance of a test stimulus is just noticeable from the surrounding luminance. The noise in the dark areas is less perceptible than that in the regions of high luminance. Because of JND, we can discard the signal below this threshold when transform the encoded bitstream. So we can decrease the amount of data. On the other hand, we can put some special signal like watermarking in the bitstream that will not be detectable.

The JND profile of a still image is a function of local signal properties, such as background luminance, activity of luminance changes and dominant spatial frequency. JND is defined below [28]:

$$JND_S(x,y) = \max\{f_1(mg(x,y)), f_2(bg(x,y))\}, 0 \le x < H, 0 \le y < W, \quad (31)$$

where $H$ and $W$ denote the height and width of the still image. $f_1$ represents the error visibility threshold due to texture masking and $f_2$ represents the error visibility threshold due to average background luminance. $mg(x, y)$ denotes the maximal weighted average of luminance gradients around the pixel at location $(x, y)$ and $bg(x, y)$ is the average background luminance around the pixel at location $(x, y)$.

$mg(x, y)$ of the pixel at $(x, y)$ is determined by calculating the weighted average of luminance changes around the pixel in four directions [29], as shown as follows:

$$mg(x,y) = \max_{k=1,2,3,4}\{|grad_k(x,y)|\}, \quad (32)$$

and

$$grad_k(x,y) = \frac{1}{16}\sum_{i=1}^{5}\sum_{j=1}^{5} p(x-3+i, y-3+j)\cdot G_k(x,y), 0 \le x < H, 0 \le y < W, \quad (33)$$

where $p(x, y)$ denotes the pixel at $(x, y)$. Four operations, $G_k(i, j)$ for $k$ = 1,2,3,4 and $i, j$ = 1,2,3,4,5 are shown in Figure 4-4 [29].

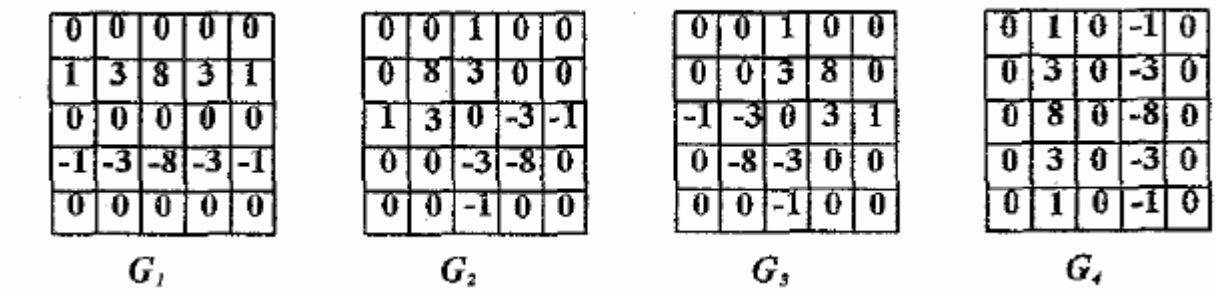



Figure 4-4 Operations for calculating the weighted average of luminance changes in four directions.

The value of $f_1(mg(x, y))$ is calculated as shown below:

$$f_1(mg(x,y)) = mg(x,y) \times \beta, 0 \le x < H, 0 \le y < W \,, \tag{34}$$

where the value of $\beta$ is get from a subject test and the value is 2/17.

*bg(x, y)* of the pixel at *(x, y)* is calculated by a weighted low-pass operator, *B(i, j), i, j* = 1,2,3,4,5, as that shown in Figure 4-5 [29]. *bg(x, y)* is calculated by:

$$bg(x,y) = \frac{1}{32}\sum_{i=1}^{5}\sum_{j=1}^{5} p(x-3+i, y-3+j) \cdot B(i,j), 0 \le x < H, 0 \le y < W \,. \tag{35}$$

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 2 | 2 | 2 | 1 |
| 1 | 2 | 0 | 2 | 1 |
| 1 | 2 | 2 | 2 | 1 |
| 1 | 1 | 1 | 1 | 1 |

***B***

Figure 4-5 The operator for calculating the average background luminance.

The relationship of between visibility threshold and the average background *bg(x, y)* is shown in Figure 4-6 [28].

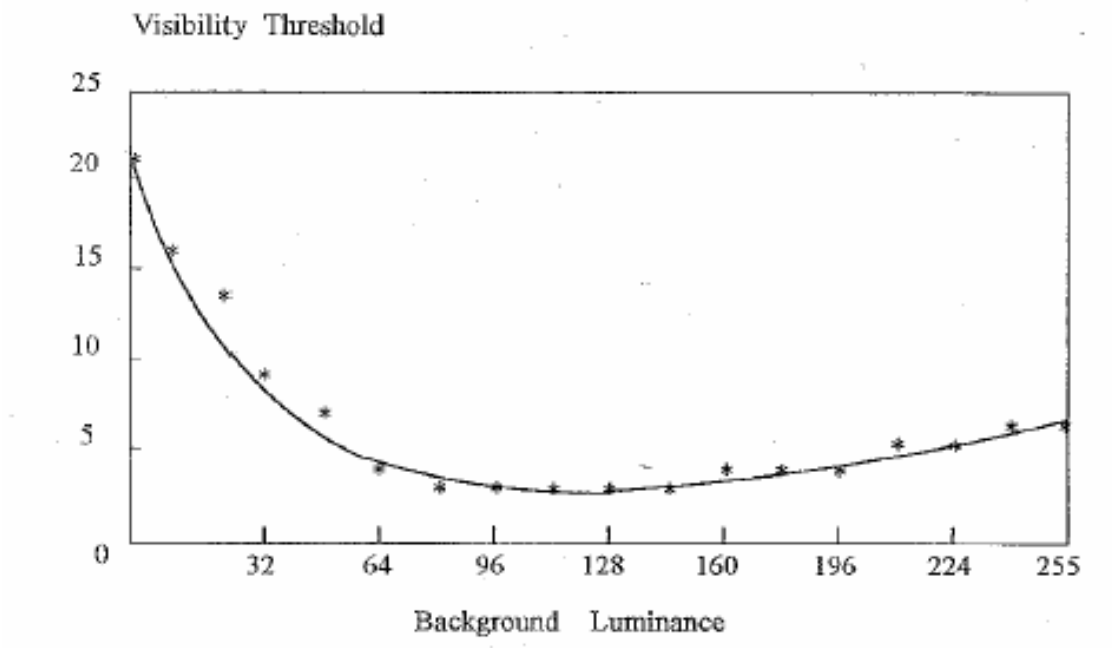



Figure 4-6 Error visibility thresholds due to background luminance in the spatial domain [28].

Sometimes we want to get the JND on the spatial-temporal domain. We can

simplify the process to get this value by multiply spatial JND and temporal JND, as that shown below [28]:

$$JND_{S-T}(x,y,n) = f_3(ild(x,y,n)) \cdot JND_S(x,y,n), \tag{36}$$

where *ild(x, y, n)* is the average interframe luminance difference between the *n*th and (*n*-1)th frame at pixel *(x, y)*, as shown below:

$$ild(x,y,n) = \frac{p(x,y,n) - p(x,y,n-1) + bg(x,y,n) - bg(x,y,n-1)}{2} \tag{37}$$

The empirical results of $f_3$ for all possible interframe luminance difference are shown in Figure 4-7 [28].

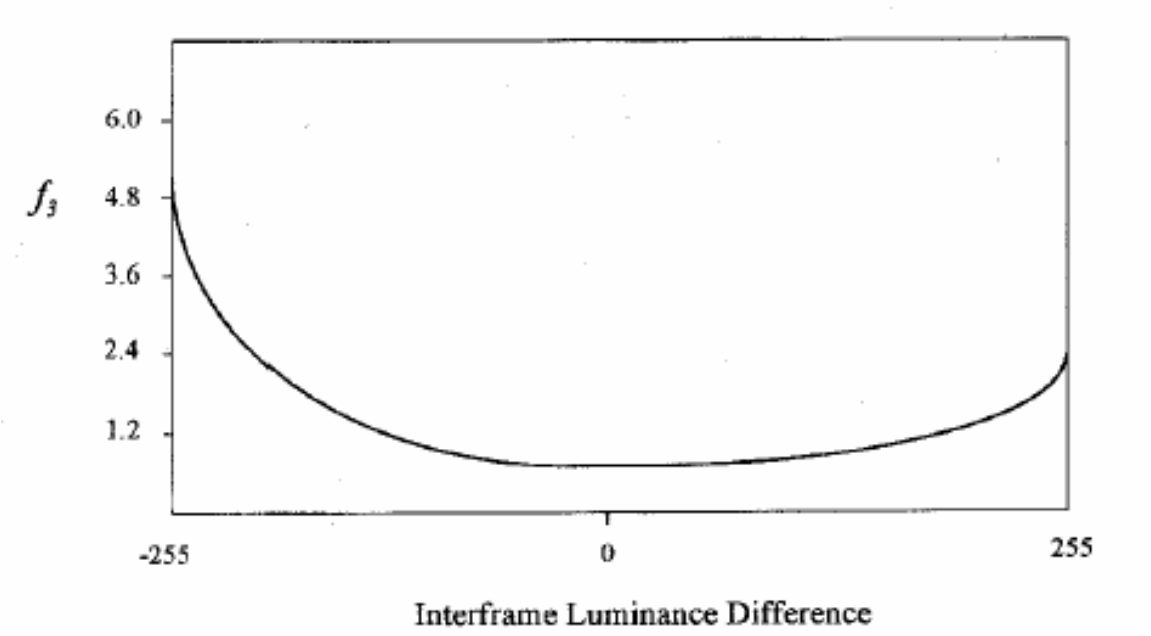



Figure 4-7 Error visibility threshold in the spatial-temporal domain, which is modeled as a scale factor or interframe luminance difference and the JND value in the spatial domain [28].

It can be seen that the error visibility threshold increases with the increasing interframe luminance difference. This coincides with the temporal masking effect that the sensitivity of human vision is decreased after scene change and large temporal luminance difference.

# Chapter 5
# Rate Control Algorithm Based on HVS

## 5.1 Transform R-D Slope Representation

The R-D slope of the truncation point *j* in the code block *i* is usually represented in the value of $\frac{\Delta D_i^j}{\Delta R_i^j} = \frac{D_i^{j-1} - D_i^j}{R_i^j - R_i^{j-1}}$ where $R_i^j$ is the accumulative bit length of truncation point *j* in code block *i* and $D_i^j$ is the accumulative distortion of truncation point *j* in code block *i*. Generally speaking, the value of $\frac{\Delta D_i^j}{\Delta R_i^j}$ is very large and the difference of this value at each truncation point is very large too.

We can transform the R-D slope of each truncation point to another representation type but keep their relative orders the same. We transform the value of $\frac{\Delta D_i^j}{\Delta R_i^j}$ to an exponential representation and use the exponent as the new R-D slope value of each truncation point, as shown in equation (38).

$$\left\lfloor 0.5 + 2^9 * \log_2\left(\frac{\Delta D_i^j}{\Delta R_i^j}\right) \right\rfloor \tag{38}$$

The new R-D slope of each truncation point is smaller and the relative difference of them is smaller too. The most important thing is that the relative order of the new R-D slopes of truncation points is kept the same as the original R-D slopes. We use this new value as the R-D slope value for each truncation point and do rate control on this new R-D slope.

## 5.2 Weighting Factor

Human vision has different sensitivity on different spatial frequency, so we need to have higher fidelity on the low spatial frequency data, which has higher sensitivity and lower fidelity on the high spatial frequency data, which has lower sensitivity. For this reason, we can convert the mean-squared error (mse) distortion to the "visual distortion" in doing rate control. In other words, we can multiply the R-D slope of each truncation point by a weighting factor such that the value of weighted R-D slope is proportional to the importance to human vision. The target is that if we use the new R-D slope value to do rate control, we can probably achieve higher visual quality. Here, we present a weighting factor only for the Y component of each frame.

Discrete wavelet transform can decompose a frame into different spatial subbands. Every subband has its own minimum visibility threshold and thus its own relative visual importance. For this reason, the weighting factor $w$ can be decomposed into two weighting factors and they are intra-subband weighting factor $w_1$ and inter-subband weighting factor $w_2$. The weighting factor $w$ is:

$$w = w_1 * w_2 \tag{39}$$

### 5.2.1 Intra-Subband Weighting Factor

The intra-subband weighting factor $w_1$ is used to decide the visibility of the truncation point in the same spatial subband. It does not consider the visibility of the truncation point in the other spatial subbands. To find the visibility of the error of a truncation point, we need to know the just-noticeable-distortion (JND) of that subband.

Watson gives the minimum threshold of luminance of each spatial subbands without masking effect [30]. This minimum threshold can be used only on the Y

component of the image. The minimum threshold $y$ of luminance of each subbands is given by [30]:

$$\log(y) = \log(a) + k \cdot (\log(f) - \log(g_\theta f_0))^2, \tag{40}$$

where the value of $a$ is 0.495, $k$ is 0.466, and $f_0$ is 0.401. The value of $g_\theta$ is 1.501, 1, and 0.534 for LL, LH/HL, and HH subbands. $f$ is spatial frequency and the value is different for different viewing condition. Under the computer monitor viewing condition, the display resolution $r$ is 16 pixels/degree.

The size of our test sequence is 288 pixels in height and 352 pixels in width. The viewing distance is about 3.5 times of the height, i.e., 1000 pixels. The visual angle in height of this condition is $2*\tan^{-1}(288/(1000*2))$ = 16.38 degree. The display resolution in height is 288/16 = 17.58 pixels/degree. The visual angle in width of this condition is $2*\tan^{-1}(352/(1000*2))$ = 19.96 degree. The display resolution in height is 352/20 = 17.6 pixels/degree. So the display resolution $r$ is about 16 pixels/degree.

The spatial frequency of each DWT level λ is $f = r * 2^{-\lambda}$ cycles/degree. Figure 5-1 shows a frame after three level of DWT and the spatial frequency of each subbands. It also shows the minimum threshold $y$ calculated by equation (40) when the maximum spatial frequency is 16.0 cycles/degree without masking effect of each subbands.

We conclude the step of calculating the minimum threshold $y$ as follows.

1) Find out the corresponding spatial frequency of each level λ by $f = r * 2^{-\lambda}$.
2) Find out the corresponding value of $g_\theta$ of each corresponding orientation.
3) Use equation (40) to calculate the minimum threshold $y$ of each subband.

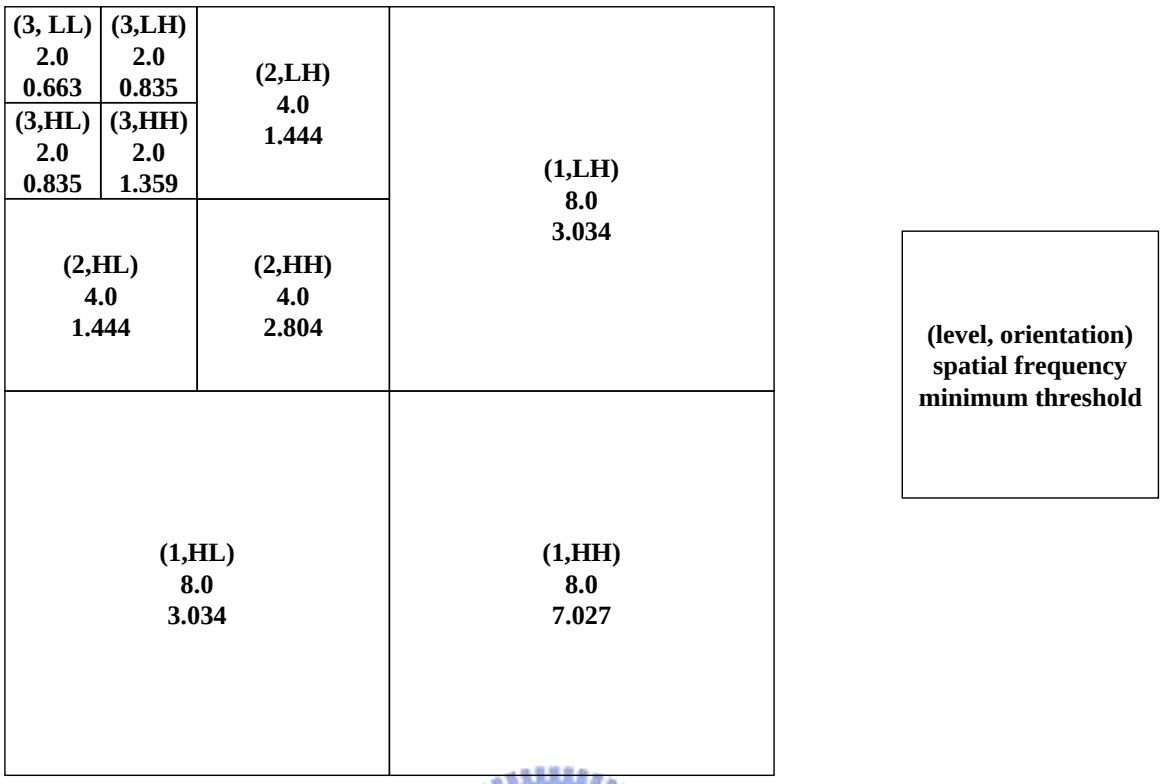



Figure 5-1 The level, orientation, spatial frequency, and minimum threshold of each DWT subbands.

After we get the minimum threshold of each subband, we need to consider the contrast masking effect of each subband. Peli proposed a definition of contrast that can be used in complex images [22], as shown in equation (26). The problem now is the contrast sensitivity for each subband. If we assume the local luminance to be constant across the whole image and equal to the average value of the coefficients in the lowest spatial subband [31], we can calculate the contrast at each location *(i, j)* in the frame in a simplified way by:

$$c(i,j) = \frac{C(i,j)}{E(C_{lowest-spatial-subband})}, \tag{41}$$

where $E(C_{lowest-spatial-subband})$ is the average of the coefficients in the lowest spatial subband and *C(i, j)* is the associated wavelet coefficient at location *(i, j)*. In the case shown in Figure 5-1, $E(C_{lowest-spatial-subband})$ is the average of the coefficients in the

subband (3, LL). Then, *c(i, j)* is the contrast of the location *(i, j)* in the frame.

The visibility of a signal can be reduced by the presence of another signal, i.e., the contrast masking effect. The masking function is shown in Figure 5-2 and it can be the same for every subband [32].

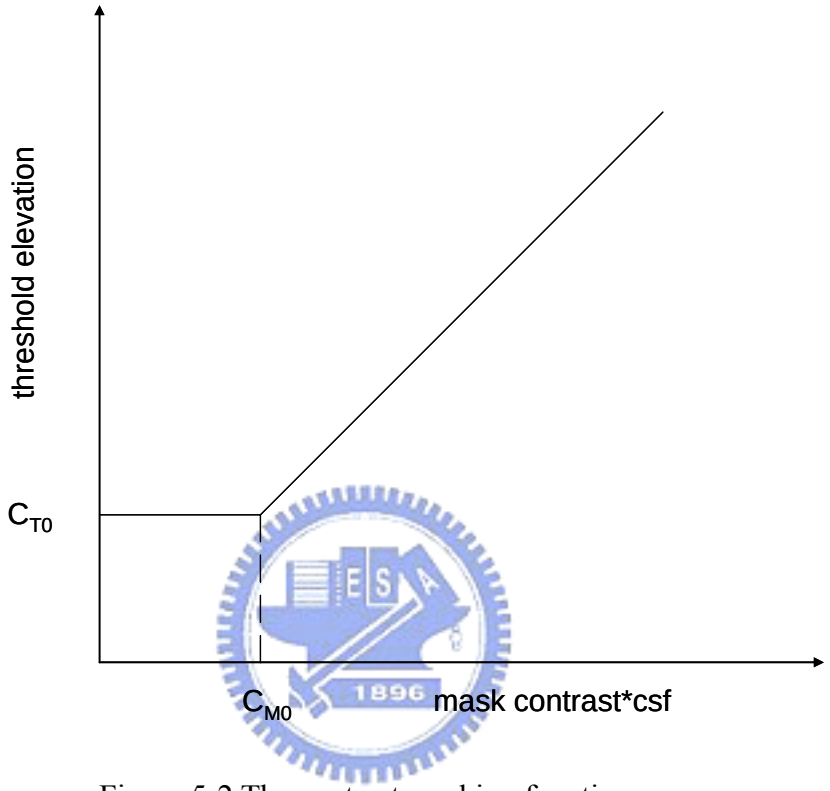



Figure 5-2 The contrast masking function.

The contrast masking function can be formulated by:

$$C_T(C_M) = C_{T0}, \qquad if\ C_M < C_{M0}, \tag{42}$$

and

$$C_T(C_M) = C_{T0}(C_M / C_{M0})^{\varepsilon}, \tag{43}$$

where $C_M$ is the masking contrast value, $C_T$ is the threshold elevation value, $\varepsilon$ is the slope. We can see that the contrast masking function is divided into a threshold range, where the target detection threshold is independent of the masking contrast, and a masking range, where it grows with the power of the masking contrast. The slope $\varepsilon$ is one for all subbands, which corresponds to experimentally derived slopes for phase-incoherent (noise) masking [32]. We generally assume that $C_{T0} = C_{M0}$ [33] and

it is confirmed by the experiments [34]. The values of $C_{T0}$ and $C_{M0}$ are all 1 [32].

If we normalize both the test threshold and masking contrast axes by the test frequency's threshold in a uniform field (i.e., 1/csf(f)), Figure 5-2 can be used to describe all frequencies, provided the test signal and masking signal are the same frequency [32]. The relationship between the threshold elevation $C_T(f, C_M)$ and real threshold value $T(f, C(f))$ is [32]:

$$C_T(f, C_M) = T(f, C(f)) \cdot csf(f) = T(f, C(f)) / T(f, 0), \tag{44}$$

where $f$ is the spatial frequency [32]. Then, the relationship between the real masking contrast value $C(f)$ and the masking contrast value $C_M$ is:

$$C_M = C(f) \cdot csf(f), \tag{45}$$

We can see that when there is no masking contrast effect, the minimum value of real threshold value $T(f, 0)$ is the inverse value of the corresponding contrast sensitivity function. We can get real threshold value $T(f, C(f))$ by dividing threshold elevation value $C_T(f, C_M)$ by corresponding contrast sensitivity value $csf(f)$ and it equals to the value $y$ we get from equation (40) when $C(f)$ is 0, i.e., no masking effect. The real masking contrast value $C(f)$ of location $(i, j)$ in the frame equals the value $c(i, j)$ we get from equation (41). We can see that the minimum real threshold values of the pixels within the same spatial subband are all the same and equal to $T(f, 0)$. Because of the different real masking contrast value $C(f)$ at different pixel, each pixel may have its own real threshold value $T(f, C(f))$.

We can use the contrast masking function to find out the corresponding threshold value of each location $(i, j)$ in a frame. Thus, we can find out the real threshold value of every pixel within the same subband and choose the smallest real threshold value as the real threshold value of the subband. But if there is one value has smallest real threshold value, i.e., $T(f, 0)$, then we need to choose this value as the real threshold

value of this subband and the masking contrast effect is of no use.

We have done some experiments, i.e., we use DWT to decompose the Y component of a frame and use different quantization step sizes to quantize one subband without quantizing the other subbands. Then, we use IDWT to reconstruct the frame and see which size of the quantization step size will produce difference between the original and the reconstructed frame that can be detected by eyes. We found that the step size we get is usually larger than the value calculated by the methods described in the above, especially for the lower spatial frequency subbands. The reason is that there may be some pixel in a subband has minimum real threshold value *T(f, 0)*, but it does not dominate the entire visual effect. For this reason, we choose the middle value of the real threshold value $T(C_{middle})$ of pixels within the same subband as the real threshold value of this subband. *T(C)* is the real threshold value of the pixel with real masking contrast value *C* and $C_{middle}$ is the pixel has middle real masking contrast value among the pixels within the same subband. In other words, $T(C_{middle})$ is also the middle real threshold value among the pixels within the same subband.

In order to apply the real threshold values to HVS, we need to convert the real threshold values from the spatial domain to the wavelet domain. We need to estimate the size of the wavelet coefficient of each subband that produces the detectable spatial (impulse) response. To do this, we have a "worst case" formula that estimates the minimum coefficients detection threshold $t_{JND}(\lambda,\theta,C)$ of the corresponding subband with level λ and orientation θ that can produce the detectable spatial response [31]:

$$t_{JND}(\lambda,\theta,C) = \frac{T(C)}{i_\theta \cdot p_l^{2(\lambda-1)}}, \tag{46}$$

where *T(C)* is the real threshold value of the corresponding subband obtained in the above and $i_\theta$ is either $p_l^2$, $p_h^2$, or $p_l p_h$ for the LL, HH, or LH/HL subbands,

respectively. $p_l$ is the maximum coefficient amplitude of the low pass synthesis filter and $p_h$ is the maximum coefficient amplitude of the high pass synthesis filter. The DWT filer we used is Daubechies 9/7 filter and the synthesis filter coefficients are shown in Table 5-1 [35]. We can see that $p_l$ is 1.115087052456994 and $p_h$ is 0.6029490182363579.

| index | Synthesis low pass filter | Synthesis high pass filter |
|---|---|---|
| 0 | 1.115087052456994 | 0.6029490182363579 |
| ±1 | 0.5912717631142470 | -0.2668641184428723 |
| ±2 | -0.05754352622849957 | -0.07822326652898785 |
| ±3 | -0.09127176311424948 | 0.01686411844287495 |
| ±4 | | 0.02674875741080976 |

Table 5-1 The coefficients of the Daubechies 9/7 synthesis filters.

We use equation (46) to calculate $t_{JND}(\lambda,\theta,C)$ of the decomposed subbands shown in Figure 5-1 and show the result in Figure 5-3. Please note that $t_{JND}(\lambda,\theta,C)$ shown in Figure 5-3 is calculated without contrast masking effect. It means that it equals to $t_{JND}(\lambda,\theta,0)$.

$t_{JND}(\lambda,\theta,0)$ is also the JND threshold of the corresponding subband, i.e., the maximum error that can be tolerated in the subband without considering masking effect. For uniform quantization, if the step size of the quantizer is *Q*, then the maximum possible error is *Q*/2 [30]. Thus we can use the quantizer with step size $2*t_{JND}(\lambda,\theta,0)$ to quantize the corresponding subband, thus the reconstructed frame will not be distinguished from the original frame by human vision.

We choose $t_{JND}(\lambda,\theta,C_{middle})$ as the minimum coefficients detection threshold for the corresponding subband.

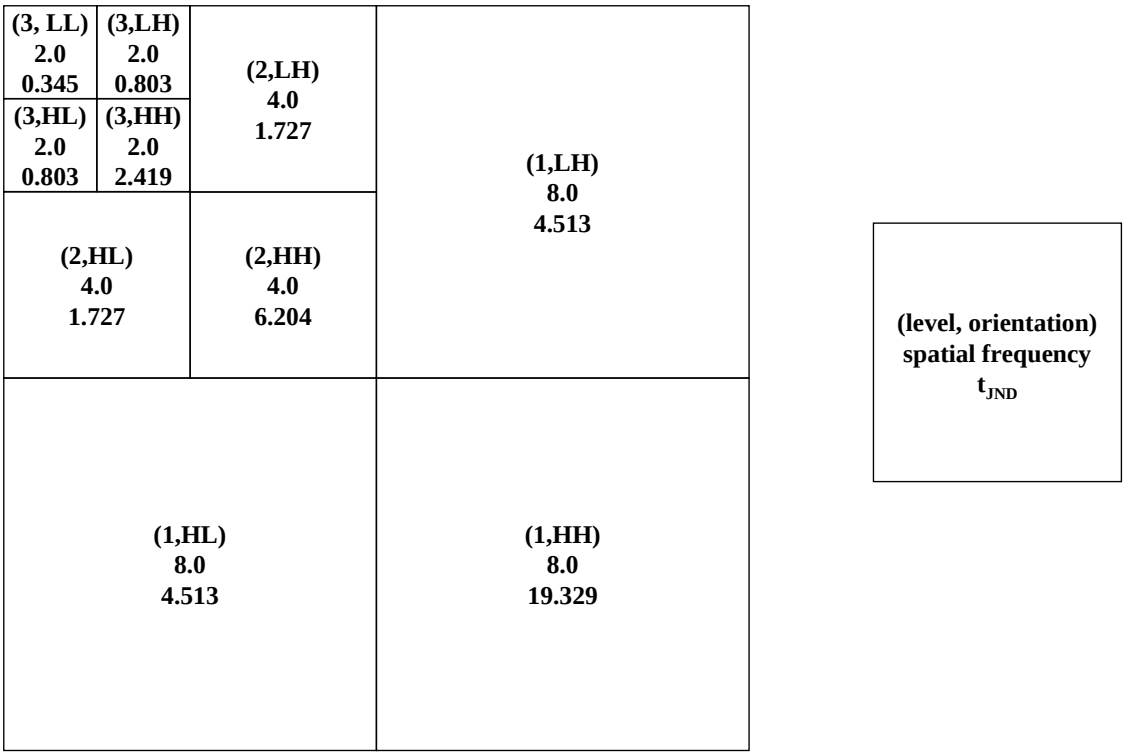



Figure 5-3 $t_{JND}(\lambda,\theta,0)$ of the frame shown in Figure 5-1.

While the $t_{JND}(\lambda,\theta,C_{middle})$ of each subband is obtained, a perceptual distortion metric that also accounts for the spatial and spectral summation of individual quantization errors is needed. The probability summation model is adopted in the perceptual distortion metric [36] [37]. The probability summation model considers a set of independent detectors, one at subband location $(\lambda,\theta,x,y)$ [37]. $(\lambda,\theta,x,y)$ is the location *(x, y)* within the subband corresponding to level λ and orientation θ. The probability of detecting a distortion at location $(\lambda,\theta,x,y)$ is determined by the psychometric function, as shown below [37]:

$$p_{(\lambda,\theta,x,y)} = 1 - \exp\left(-\left|\frac{e(\lambda,\theta,x,y)}{t_{JND}(\lambda,\theta,x,y)}\right|^{\beta_b}\right), \tag{47}$$

where $e(\lambda,\theta,x,y)$ is the quantization error at location $(\lambda,\theta,x,y)$ and $\beta_b$ is a parameter whose value is chosen to achieve consistency between (39) and the experimentally determined psychometric function for a given type of distortion. We

choose the value of $\beta_b$ is 4 [36] [37]. $t_{JND}(\lambda,\theta,x,y)$ is the minimum threshold value of location $(\lambda,\theta,x,y)$, but we set the minimum threshold value of all the coefficients within the same subband are the same and equals to $t_{JND}(\lambda,\theta,C_{middle})$. Thus, we use $t_{JND}(\lambda,\theta,C_{middle})$ to replace $t_{JND}(\lambda,\theta,x,y)$, as shown below:

$$p_{(\lambda,\theta,x,y)} = 1 - \exp(-\left|\frac{e(\lambda,\theta,x,y)}{t_{JND}(\lambda,\theta,C_{middle})}\right|^4). \tag{48}$$

The highest visual acuity is limited to the size of the foveal region and covers approximately $2^\circ$ of visual angle in HVS. Let $F_{(n1,n2)}$ denote the area in the spatial domain that is centered at location *(n1, n2)* and covers $2^\circ$ of visual angle. Then, the probability $P_{F_{(n1,n2)}}$ of detecting a distortion in this region is [37]:

$$P_{F_{(n1,n2)}} = 1 - \prod_{(\lambda,\theta,x,y)\in F}(1 - p_{(\lambda,\theta,x,y)}). \tag{49}$$

The probability summation scheme is developed based on two assumptions [36] [37].

1) A distortion is detected in the foveal region if and only if at least one detector signals the presence of distortion.

2) The probability of detecting a distortion of each detector is independent.

We can substitute equation (48) into (49), thus we have [37]:

$$P_{F_{(n1,n2)}} = 1 - \exp(-(D_{F_{(n1,n2)}})^4), \tag{50}$$

where

$$D_{F_{(n1,n2)}} = (\sum_{(\lambda,\theta,x,y)\in F}\left|\frac{e(\lambda,\theta,x,y)}{t_{JND}(\lambda,\theta,C_{middle})}\right|^4)^{\frac{1}{4}} = (\frac{\sum_{(\lambda,\theta,x,y)\in F}\left|e(\lambda,\theta,x,y)\right|^4}{(t_{JND}(\lambda,\theta,C_{middle}))^4})^{\frac{1}{4}}. \tag{51}$$

The maximum width, maximum height, and maximum depth of the code block in 3D-ESCOT coding are 64, 64, and 4. Because we only consider one frame each time and $t_{JND}(\lambda,\theta,C_{middle})$ of different frame may not be the same, the depth of the code block is 1. Although human eyes can see the scenery in the visual angle about $160^\circ$

to $180^{o}$, human can only pay attention to the scenery in the visual angle about $2^{o}$ because of the structure of the fovea. If we assume the foveal region is the code block, the maximum visual angle of each code block is $4^{o}$ in our condition. So we need to modify equation (51) to fit it to our condition.

From equation (51), we can see that the total "visual error distortion" is $\sum_{(\lambda,\theta,x,y)\in F}\left|e(\lambda,\theta,x,y)\right|^4 = \sum_{x=0}^{block\_width-1}\sum_{y=0}^{block\_height-1}\left|e(\lambda,\theta,x,y)\right|^4$ and the total "visual error distortion" that can be tolerated is $block\_height*block\_width*t_{JND}(\lambda,\theta,C_{middle})^4$. We think that the ratio of these two values can determine the visual error probability. So we rewrite equation (51) into:

$$D_{(\lambda,\theta)} = \left(\frac{\sum_{x=0}^{block\_width-1}\sum_{y=0}^{block\_height-1}\left|e(\lambda,\theta,x,y)\right|^4}{block\_height*block\_width*(t_{JND}(\lambda,\theta,C_{middle}))^4}\right)^{\frac{1}{4}}. \tag{52}$$

The spatial subband may include more than one code block and each code block has its own height and width. If we consider just one code block a time, we can get:

$$D_{(\lambda,\theta,z)} = \left(\frac{\sum_{x=0}^{W(\lambda,\theta,z)-1}\sum_{y=0}^{H(\lambda,\theta,z)-1}\left|e(\lambda,\theta,z,x,y)\right|^4}{H(\lambda,\theta,z)*W(\lambda,\theta,z)*(t_{JND}(\lambda,\theta,C_{middle}))^4}\right)^{\frac{1}{4}}, \tag{53}$$

where $H(\lambda,\theta,z)$ and $W(\lambda,\theta,z)$ represents the height and width of the *z*-th code block in spatial subband $(\lambda,\theta)$ and $e(\lambda,\theta,z,x,y)$ is the error in the location *(x, y)* of the *z*-th code block in spatial subband $(\lambda,\theta)$.

We can combine equation (50) and (53) together, then we can get the intra-subband weighting factor $w_1$ of the coding pass of the corresponding bitplane:

$$w_1 = 1-\exp\left(-\left(\frac{\sum_{x=0}^{W(\lambda,\theta,z)-1}\sum_{y=0}^{H(\lambda,\theta,z)-1}\left|e(\lambda,\theta,z,x,y)\right|^4}{H(\lambda,\theta,z)*W(\lambda,\theta,z)*(t_{JND}(\lambda,\theta,C_{middle}))^4}\right)\right). \tag{54}$$

and $e(\lambda,\theta,z,x,y)$ is the total error of the coding pass of the corresponding bitplane.

We can see that intra-subband weighting factor $w_1$ is different for every truncation

point even the truncation points are located in the same spatial subband, and $w_1$ is frame-dependent.

### 5.2.2 Inter-Subband Weighting Factor

Intra-subband weighting factor $w_1$ is close to 1 when the bitplane is close to the most significant bitplane (large distortion). In other words, if we multiply the R-D slope of each truncation point by $w_1$, the R-D slope of bitplane near the most significant bitplane may not change and the R-D slope of bitpane near the least significant bitplane becomes smaller. Thus, the visual quality is the same as that in original rate control algorithm at low bit rate. This means that we need to find out another weighting factor to decide the relative visual importance of the same bitplane in different spatial subbands. This is inter-subband weighting factor $w_2$.

From equations (27), (28), (29), and (30), we can see that the sensitivity at different spatial frequency is very different. Thus, the difference between their associated inter-subband weighting factors should be large too. But we use equation (38) to represent the R-D slope of the truncation point, the relative difference between their inter-subband weighting factors becomes smaller too.

We use $t_{JND}(\lambda,\theta,0)$ instead of $t_{JND}(\lambda,\theta,C_{middle})$ to calculate $w_2$. The reason is that the spatial subband with lower $t_{JND}(\lambda,\theta,0)$ usually has higher sensitivity. If we consider masking contrast effect, we can get $t_{JND}(\lambda,\theta,C_{middle})$ and $t_{JND}(\lambda,\theta,C_{middle})$ is bigger than or equal to $t_{JND}(\lambda,\theta,0)$. Thus, the associated intra-subband weighting factor $w_2$ will be smaller.

The $t_{JND}(\lambda,\theta,0)$ of the lowest spatial subband is the smallest of all the subbands but its $t_{JND}(\lambda,\theta,C_{middle})$ is usually very large because of large contrast masking effect due to large wavelet coefficients in this subband. If we use $t_{JND}(\lambda,\theta,C_{middle})$

to calculated $w_2$, we may think that the minimum spatial subband has lower $w_2$. It is not the true based on our experiments. From our experiments, we found that the lowest spatial subband has the largest weighting. For this reason, we use $t_{JND}(\lambda,\theta,0)$ to calculate $w_2$ for each spatial subband.

Assuming the $t_{JND}(\lambda,\theta,0)$ of the lowest spatial subband is $t_{JND-lowest-spatial-subband}$. For the frame showing in Figure 5-3, $t_{JND-lowest-spatial-subband} = t_{JND}(3,LL,0) = 0.345$. We find that $t_{JND-lowest-spatial-subband}$ is the smallest $t_{JND}(\lambda,\theta,0)$ of all the spatial subbands. The inter-subband weighting factor $w_2$ of spatial subband $(\lambda,\theta)$ is:

$$w_2 = 1 + \frac{\exp(\frac{t_{JND-lowest-spatial-subband}}{t_{JND}(\lambda,\theta,0)})}{10} \tag{55}$$

From equation (55), we can see that the inter-subband weighting factor $w_2$ is the same for all the truncation points within the same spatial subband and it is frame-independent.

Combing equations (39), (54), and (55) together, we can get the function of subband weighting factor $w$:

$$w = w_1 * w_2 = (1 - \exp(-(\frac{\sum_{x=0}^{W(\lambda,\theta,z)-1}\sum_{y=0}^{H(\lambda,\theta,z)-1}\left|e(\lambda,\theta,z,x,y)\right|^4}{H(\lambda,\theta,z)*W(\lambda,\theta,z)*(t_{JND}(\lambda,\theta,C_{middle}))^4}))) * (1 + \frac{\exp(\frac{t_{JND-lowest-spatial-subband}}{t_{JND}(\lambda,\theta,0)})}{10}) \tag{56}$$

We can use $w$ to transform the original distortion to "visual distortion", i.e., the weighted truncation points are in the order of visual importance.

## 5.3 Rate Control

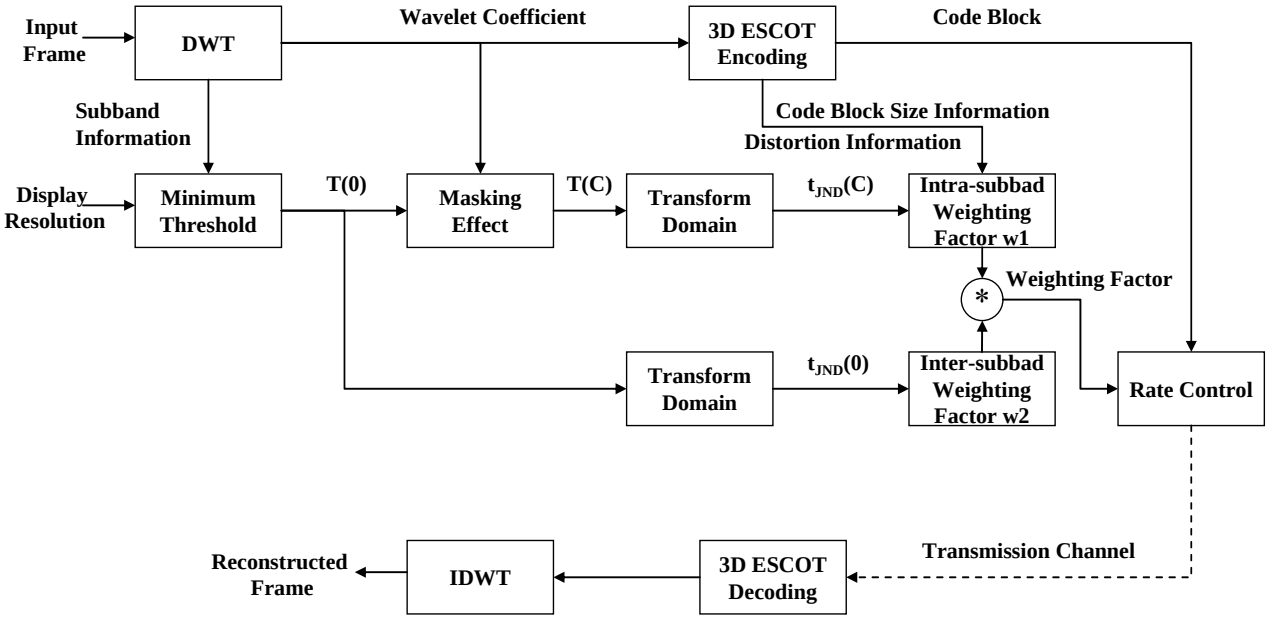



Figure 5-4 The flow chart of calculating the subband weighting factor *w*.

Figure 5-4 shows the flow chart of calculating the subband weighting factor *w*. We need to transform the R-D slope of each truncation point to new R-D slope by equation (38). Then, we can get subband weighting factor of each truncation point by equation (56) and multiply it to the new R-D slope got from equation (38).Thus, we can obtain the R-D slope value based on "visual distortion".

We use the new weighted R-D slope to do rate control. If the truncation point has larger new weighted R-D slope, it has high probability to be packaged and transmitted. We show the experimental results in the next subsection and examine the correctness of the proposed rate control algorithm.

## 5.4 Experimental Results

Here we show two types of the experimental results. One is the correctness of the proposed rate control algorithm and the other is the comparison between the original and proposed rate control algorithm.

### 5.4.1 Correctness of the Proposed Rate Control Algorithm

We propose a method to examine the correctness of the proposed rate control algorithm. We calculate the $t_{JND}(\lambda,\theta,C_{middle})$ of each spatial subband of the Y component of the frame. Then, we discard the coding pass (for 3D-ESCOT, each bitplane has 3 coding passes, except for the first bitpalne that has only 1 coding pass) of the bitplane that smaller than $2*t_{JND}(\lambda,\theta,C_{middle})$ and calculate the smallest bit rate to transmit the necessary data, i.e., we use the quantizer based on HVS to quantize the wavelet coefficients and transmit. The bit rate is calculated under the assumption that we transmit 30 frames per second. We will compare the discarded coding pass of the original and proposed rate control algorithm at the same bit rate. We only check the Y component of the first frame for several test sequences. There are 64 code blocks in a frame. We will show the original frame, the HVS quantized frame (HVS quantizer), the frame reconstructed using the Microsoft original rate control algorithm (MS original), and the frame reconstructed using the proposed rate control algorithm (weighting scheme). We will compare the number of coding passes they discard to see the difference between them.

(a)Original

(b)HVS quantizer, PSNR = 40.09dB, package size = 18702bytes

(c)MS original, PSNR = 42.08dB, package size = 18724bytes

(d)Weighting scheme, PSNR = 39.79dB, package size = 18377bytes

Figure 5-5 The four test frames for comparison of test frame I.

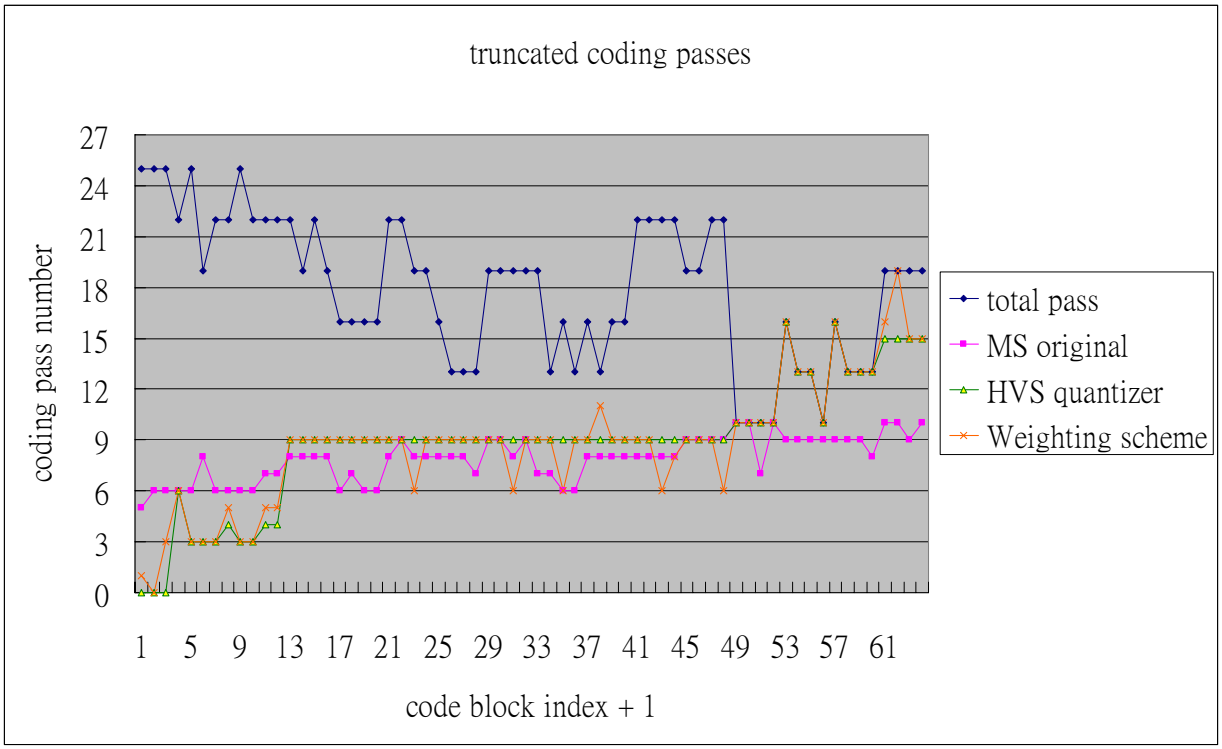



Figure 5-6 The truncated coding passes of test frame I. The required bit rate is 4.23M bytes per second if the frame rate is 30 frames/sec.

(a)Original

(b)HVS quantizer, PSNR = 39.15dB, package size = 24853bytes

(c)MS original, PSNR = 40.80dB, package size = 24820bytes

(d)Weighting scheme, PSNR = 39.01dB, package size = 24652bytes

Figure 5-7 The four test frames for comparison of test frame II.

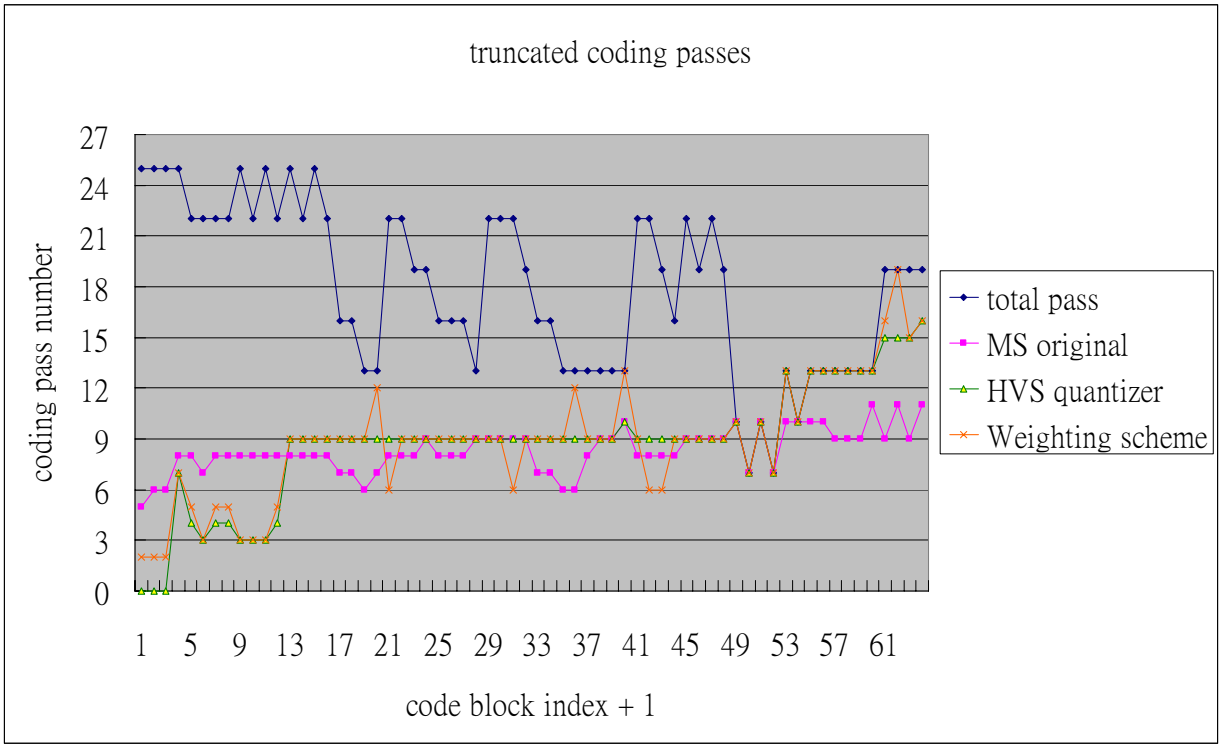
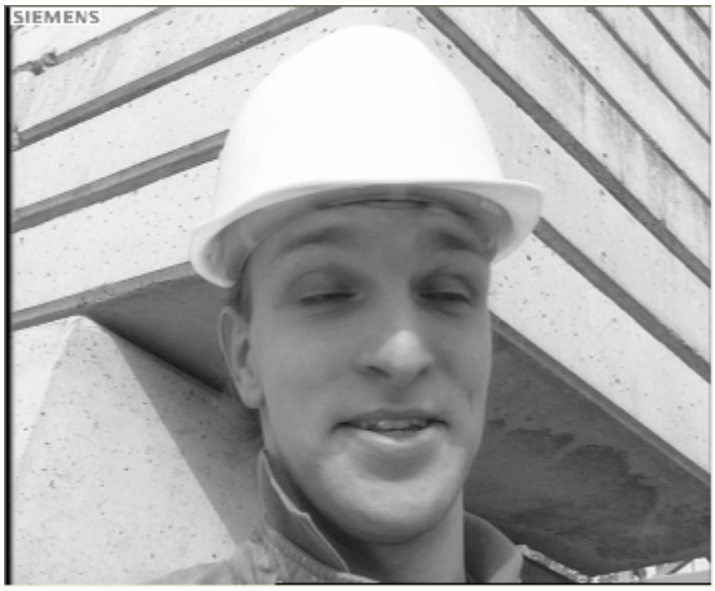
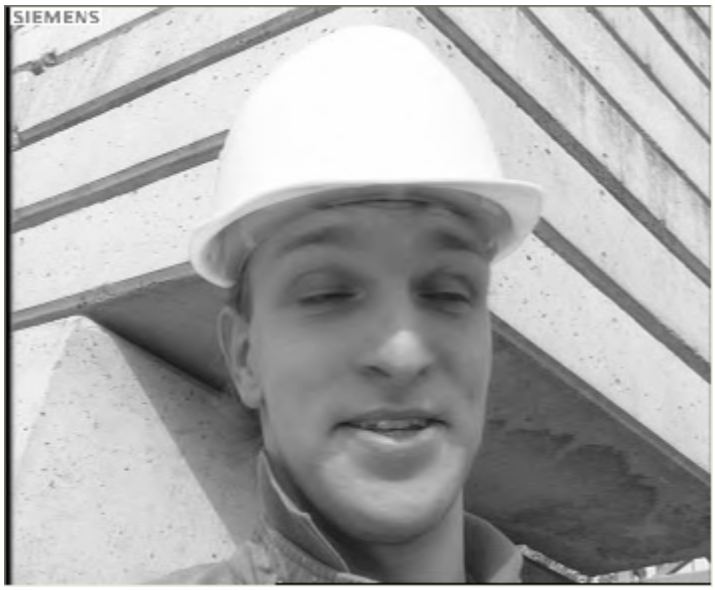



Figure 5-8 The truncated coding passes of test frame II. The required bit rate is 5.64M bytes per second if the frame rate is 30 frames/sec.

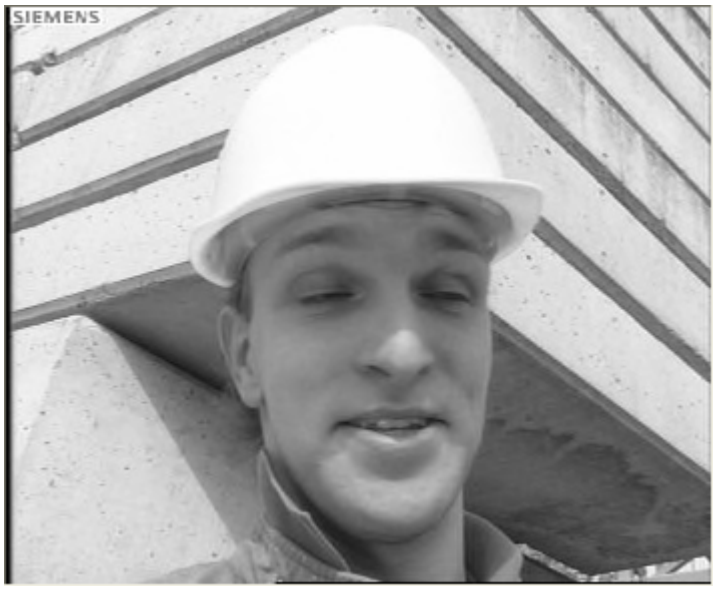



(a)Original


(b)HVS quantizer, PSNR = 40.21dB, package size = 15675bytes




(c)MS original, PSNR = 41.90dB, package size = 15695bytes




(d)Weighting scheme, PSNR = 39.97dB, package size = 15577bytes

Figure 5-9 The four test frames for comparison of test frame III.

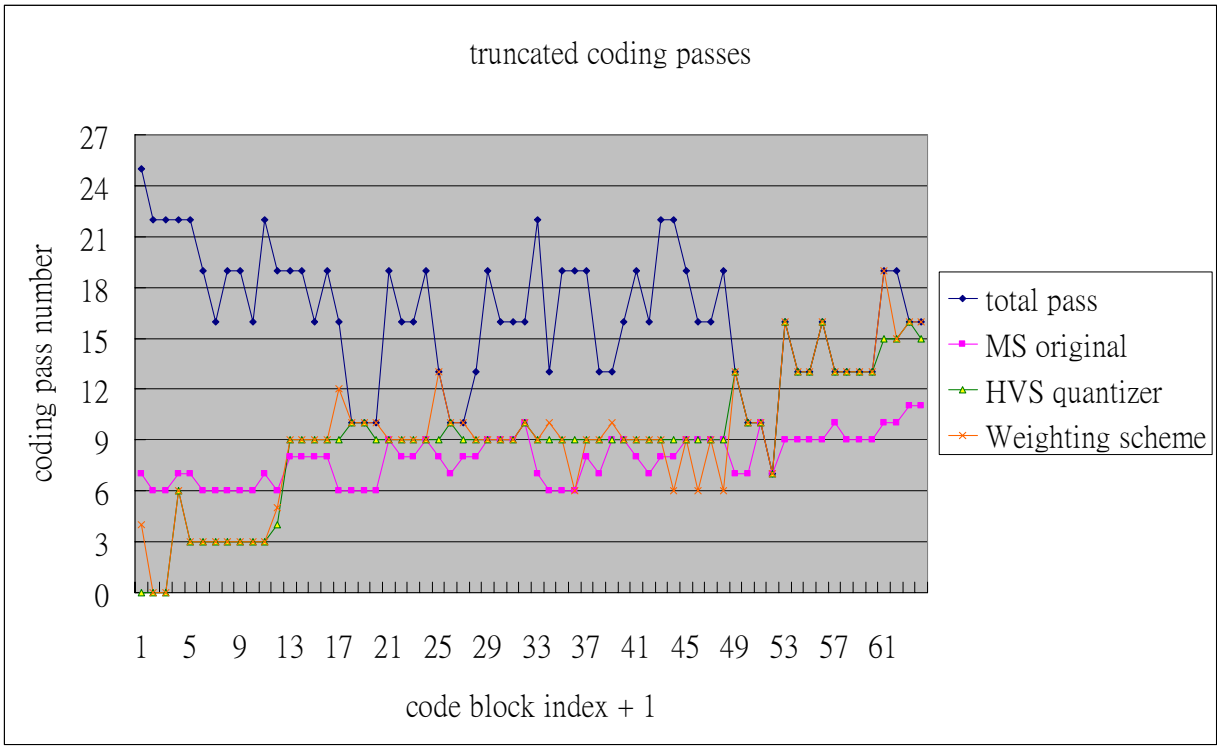



Figure 5-10 The truncated coding passes of test frame III. The required bit rate is 3.54M bytes per second if the frame rate is 30 frames/sec.


(a)Original


(b)HVS quantizer, PSNR = 39.88dB, package size = 21668bytes


(c)MS original, PSNR = 41.90dB, package size = 21570bytes


(d)Weighting scheme, PSNR = 39.40dB, package size = 21608bytes

Figure 5-11 The four test frames for comparison of test frame IV.

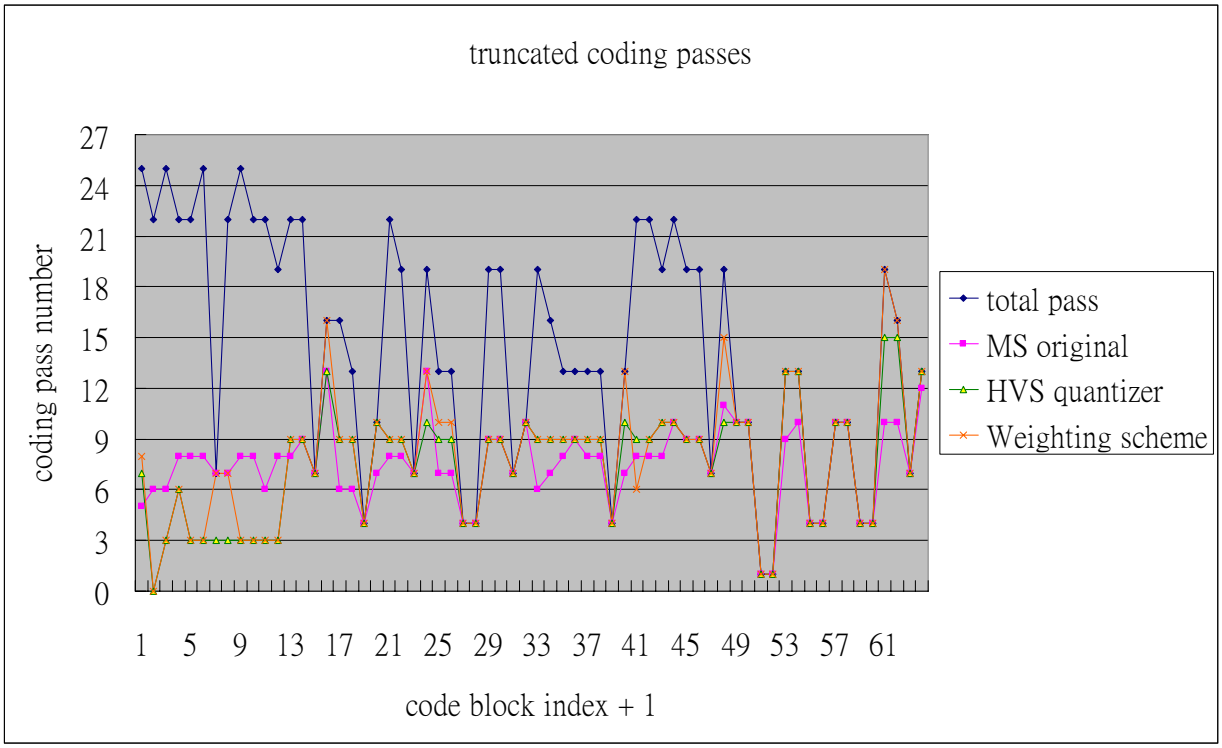



Figure 5-12 The truncated coding passes of test frame IV. The required bit rate is 4.92M bytes per second if the frame rate is 30 frames/sec.

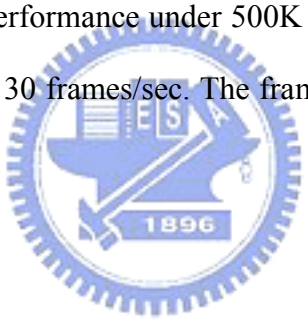
For the frames in Figure 5-5, we can not distinguish between these four frames. But the PSNR value of each of frame is different. We can find the same condition in Figure 5-7, Figure 5-9, and Figure 5-11. The quantizer that used in HVS quantizer is smaller than or equals to $2*t_{JND}(\lambda,\theta,C_{middle})$. The PSNR values of the weighting scheme are lower than those of MS original but almost equals to the PSNR values of HVS quantizer. From Figure 5-6, we can see that the numbers of truncated coding passes of HVS quantizer and weighting scheme are very similar for each coding block. We can see the same condition in Figure 5-8, Figure 5-10, and Figure 5-12. For this reason, we believe that the proposed weighting scheme is correct. The package size of weighting scheme is usually smaller than that of MS original. (In Figure 5-11, the package size of weighting scheme is larger than that of MS original.)

From above four test frames, we can see that the frame quantized by the quantizer

based on HVS is almost the same to the original frame. From the truncated coding passes of each test frame, we can see that human eyes can tolerate larger error in high spatial frequency than in low spatial frequency. We can see that the truncated coding passes of the reconstructed frames using the weighting scheme are very similar to those of the frames quantized by the quantizer based on HVS at the same bit rate. Because the bit rate for each test frame is very high, especially we only encode and transmit the Y component of the frame. We like to compare the performance on visual quality of these two rate control algorithms at lower bit rates.

### 5.4.2 Comparison of Rate Control Algorithms

Here we compare the visual quality difference between two different rate control algorithms. We compare the performance under 500K bits per second and 1000K bits per second if the frame rate is 30 frames/sec. The frames to be tested are the same as the previous section.



(a)MS original, PSNR = 27.44dB, package size = 2217bytes

(b)Weighting scheme, PSNR = 27.13dB, package size = 2185bytes

(c)MS original, PSNR = 31.19dB, package size = 4371bytes

(d)Weighting scheme, PSNR = 30.98dB, package size = 4317bytes

Figure 5-13 The four test frames of frame I at low bit rates. (a) and (b) are 500K bits per second. (c) and (d) are 1000K bits per second.

In Figure 5-13, we can see that the ocean of weighting scheme looks smoother than that of MS original. But the PSNR and the package size of the weighting scheme are all smaller than those of MS original.

(a)MS original, PSNR = 24.71dB, package size = 2216bytes

(b)Weighting scheme, PSNR = 24.43dB, package size = 2209bytes

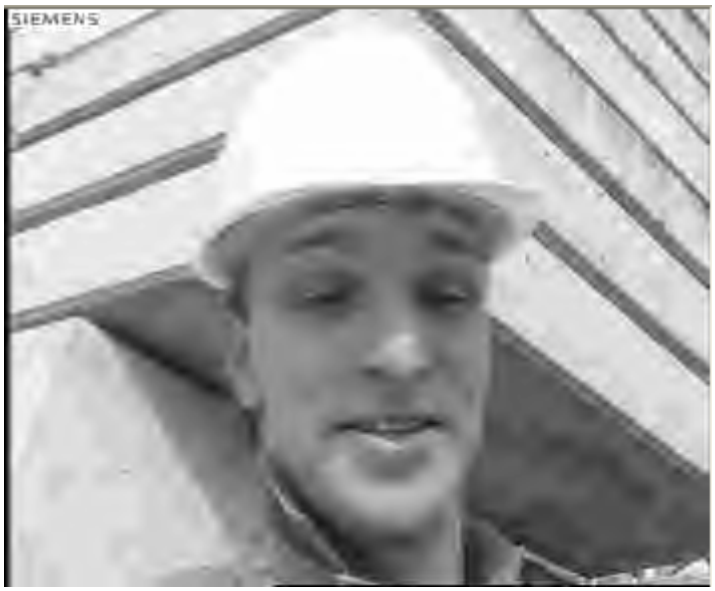
(c)MS original, 27.61dB, package size = 4368bytes

(d)Weighting scheme, 27.40dB, package size = 4383bytes

Figure 5-14 The four test frames of frame II at low bit rates. (a) and (b) are 500K bits per second. (c) and (d) are 1000K bits per second.

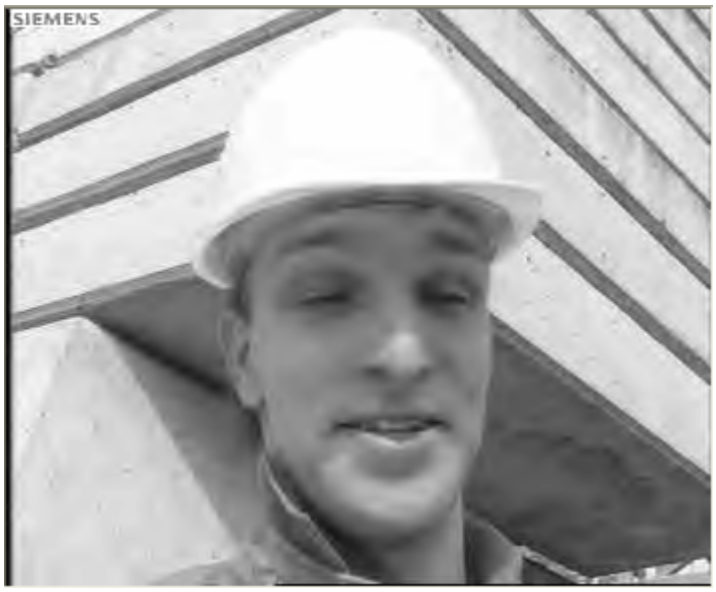
From Figure 5-14, we can see that the visual quality of weighting scheme is higher than that of MS original. The background of the weighting scheme looks smoother than that of the MS original. The PSNR of the weighting scheme is lower than that of

MS original.

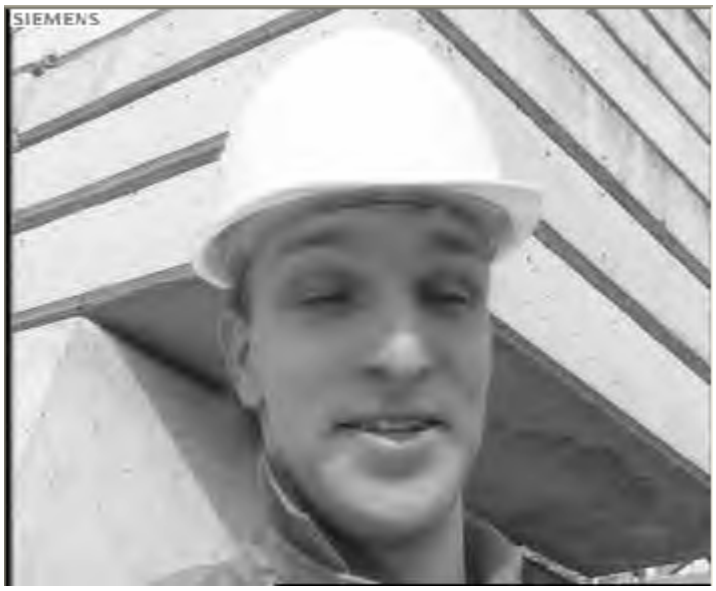



(a)MS original, PSNR = 30.44dB, package size = 2203bytes


(b)Weighting scheme, PSNR = 29.91dB, package size = 2197bytes




(c)MS original, PSNR = 34.02dB, package size = 4400bytes




(d)Weighting scheme, PSNR = 33.95dB, package size = 4342ytes

Figure 5-15 The four test frames of frame III at low bit rates. (a) and (b) are 500K bits per second. (c) and (d) are 1000K bits per second.

The visual quality of the frame in Figure 5-15(b) is clearly better than that of the frame in Figure 5-15(a). The wall and face looks smoother but the value of PSNR is lower. But the edge of the wall in Figure 5-15(b) is not so clear s that in Figure 5-15(a) because we truncate more signal in high spatial frequency subbed. The visual quality

of the frame in Figure 5-15(c) and that of the frame in Figure 5-15(d) is almost the same and their values of PSNR are almost the same, too. We found one thing that the visual quality of Figure 5-15(b) is better than that of Figure 5-15(a), but it does not look like the original frame. We can found some shadow regions on the wall and face in the original frame. We can also found shadow regions on the wall and face in Figure 5-15(a). The shadow regions on the wall and face in Figure 5-15(b) are not so clear.


(a)MS original, PSNR = 23.70dB, package size = 2196bytes


(b)Weighting scheme, PSNR = 23.14dB, package size = 2090bytes


(c)MS original, PSNR = 26.85dB, package size = 4287bytes


(d)Weighting scheme, PSNR = 26.67dB, package size = 4272bytes

Figure 5-16 The four test frames of frame IV at low bit rates. (a) and (b) are 500K bits per second. (c) and (d) are 1000K bits per second.

The wall and the floor of the frame in Figure 5-16(b) looks smoother than that of the frame in Figure 5-16(a). But the edge of the player and the letters on the wall of the frame in Figure 5-16(b) is not as clear as those of the frame in Figure 5-16(a). Also Figure 5-16(d) looks slightly better than Figure 5-16(c). The package size of

weighting scheme in Figure 5-16(b) is smaller than that of MS original in Figure 5-16(a). But we can see that the difference is almost 100 bytes. The reason we think is the relative difference between R-D slopes of associated truncation points becomes larger. If we want to package more data, we must use large bit rate.

## 5.5 Discussion

The proposed rate control algorithm can provide better visual quality, especially when there is a large and flat region in the test frames, such as the ocean in test frame I. But sometimes the visual quality of edges may become worse. The reason is that the visual weighting for high spatial frequency is smaller than the value it should have.

Because we use “human visual weighting error” instead of “quantization error” to do rate control, PSNR will become smaller. It proves that the frame with higher PSNR may not have higher visual quality. The weighting factor will make the relative difference between the R-D slopes of associated truncation points in MSB bitplane and those of associated truncation points in LSB bitplane larger. Thus, we need higher bit rate to package the same data.

Because the human vision has high sensitivity at low spatial frequency (flat region) than high spatial frequency (edge), the proposed rate control algorithm packages more data of low spatial frequency and less data of high spatial frequency. Thus we can make the flat region smoother and but larger error in edges. Larger error in edges will not be detected by the eyes sometimes. The PSNR values of the frames reconstructed by proposed rate control algorithm are always smaller than those of the frames reconstructed by original rate control algorithm. This proves that the frame has higher visual quality may not have higher PSNR value.

# Chapter 6
# Conclusion and Future Work

## 6.1 Conclusion

The interframe wavelet video coding is a compression technique that provides flexible and multi-purpose scalability. The single created by interframe wavelet video coding can provide rate/SNR, temporal, and spatial scalability.

The study on HVS is become more important in recent years. The data of HVS is usually obtained from experiments. Because HVS has different response under different conditions, this is hard to find out a global useful formula for CSF or JND that can be accepted extensively.

We propose a weighting factor that can be used to convert the distortion measure of a truncation points to a visual weighted one. It is the product of the intra-subband weighting factor and inter-subband weighting factor. They are summarized below.

1) *intra-subband weighting factor*: It decides the visual importance of errors within the same subbands. The error smaller the JND of the corresponding subband has lower weighting because of the less importance to HVS.
2) *inter-subband weighting factor*: It decides the visual importance of errors in different subbands. If the values of the errors in different spatial subbands are the same, they have different visual importance to HVS. The error in lower spatial subband often has higher visual importance.

## 6.2 Future Work

We notice there are a few work items can be future explored.

1) The function of the minimum threshold provided by Watson is based on 9/7 linear phase filter [30]. We may need to derive a function that corresponding to the Daubechies 9/7 filter.

2) We assume the local luminance is constant across the whole image but it is not correct. We like to find another model to estimate the local luminance. The estimation of masking effect in lower spatial subbands can be improved. The masking effect in lower spatial subbands is usually very large. If we can estimate it with higher precision, we can get better weighting factor to do rate control and decrease the probability of the occurrence of visual error.

3) The proposed rate control algorithm is applicable to the luminance component of a picture. We like to extend it to the chrominance component. Watson suggests the minimum threshold function on chrominance [30] but the experiment results shows that visual responses on chrominance for different people is very different.

4) The proposed rate control algorithm is now used only on one spatial decomposed frame. We like to extend it to temporal domain. There is no clear model of minimum temporal threshold because the human eyes may track the moving objects and the resolution of static objects can be low. Finding an adequate model for temporal human vision can be a difficult and unsolved problem.

# References


[1] W. P. Li, “Overview of Fine Granularity Scalability in MPEG-4 Video Standard”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.11, pp. 301-317, March 2001.

[2] J. -R. Ohm, “Three-dimensional subband coding with motion compensation”, *IEEE Transactions on Image Processing*, vol. 3:5, pp. 559-571, 1994.

[3] S. -T. Hsiang and J. W. Woods, “Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank”, *Signal Processing: Image Communications*, vol. 16, pp. 705–724, May 2001.

[4] D. Taubman and R. Rosenbaum, “Rate-distortion optimizes interactive browsing of JPEG2000 images”, *Image Processing*, 2003. September 2003.

[5] T. Kronander, “Motion compensated 3-dimensional wave-form image coding”, *International Conference on Acoustic, Speech, and Signal Processing*, vol. 3, pp. 1921-1924, 1989.

[6] S. T. Hsiang and J. W. Woods, “Invertible three-dimensional analysis/synthesis system for video coding with half-pixel-accurate motion compensation”, *SPIE Conference on Visual Communication and Image Processing*, vol. 3653, pp. 537-546, January 1999.

[7] B. Pesquet-Popescu, V. Bottreau, “Three-dimensional lifting schemes for motion compensated video compression”, International Conference on Acoustic, Speech, and Signal Processing, vol. 3, pp. 1793 -1796, 2001.

[8] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients”, *IEEE Transactions on Signal Processing*, vol. 41:12, pp. 657-660, December 1992.

[9] D. Taubman, “High performance scalable image compression with EBCOT”, *IEEE Transactions on Image Processing*, vol. 9:7, pp. 1158-1170, July 2000.

[10] S. T. Hsiang and J.W. Woods, “Embedded image coding using zeroblocks of subband-wavelet coefficients and context modeling”, in *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 3:5, pp. 662-665, May 2000.

[11] J. W. Woods, “AHG on Digital Cinema Video Coding Technology”, ISO/IEC/JTC1 SC29/WG11 doc. No. m7645, Pattaya, December 2001.

[12] P. S. Chen and J. W. Woods, “Comparison of MC-EZBC and H.26L TM8 on Digital Cinema Test Sequences”, ISO/IEC JTC1/SC29/WG11 doc. No. m8130, Cheju Island, March 2002.

[13] P. S. Chen and J. W. Woods, “Improved MC-EZBC with Quarter-pixel Motion Vectors”, ISO/IEC JTC1/SC29/WG11 doc. No. m8366, FairFax, VA, May 2002.

[14] S. S. Tsai, H. M. Hang, T. Chiang, “Exploration Experiments on the Temporal Scalability of Interframe Wavelet Coding”, ISO/IEC/JTC1 SC29/WG11 doc. No. m8959, Shanghai, October 2002.

[15] J. Xu et al, “3D subband video coding using Barbell lifting”, ISO/IEC JTC1/SC29/WG11, MPEG2004/M10569/S05, Munich, March 2004.

[16] J. Xu, Z. Xiong, S. Li, Y, Zhang, “Three-Dimensional Embedded Subband Coding with Optimized Truncation (3D ESCOT)”, Applied and Computational Harmonic Analysis 10, 290-315(2001), doi:10.1006/acha.2000.0345, available online at http://www.idealibrary.com .

[17] Y. Shoham, A.Gersho, “Efficient bit allocation for an arbitrary set of quantizers”, *IEEE Transactions on Acoustics Speech Signal Process*, Vol 36, No 9, pp1445-1443, September 1988.

[18] A. Aminlou, O. Fatemi, “Very Fast Bit Allocation Algorithm, Based On Simplified R-D Curve Modeling”, Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003 10th IEEE International Conference on, Volume: 1, 14-17, Pages:112 - 115 Vol.1 December 2003.

[19] A. N. Netravali and B. G. Haskell, *Digital Images: Representation and Compression*, 2nd ed., Plenum Press, ‘95.

[20] S. Winkler, “Issue in vision modeling for perceptual video quality assessment”, Signal Processing 78, pp. 231-252, 1999.

[21] N. R. Carlson, *Physiology of Behavior*, Allyn and Bacon, ‘94.

[22] E. Peli, “Contrast in complex images”, *J. Opt. Soc. Amer.* A, vol. 7, pp. 2032-2039, October 1990.

[23] J. L. Mannos and D. J. Sakrison, “The effect of a visual fidelity criterion on the encoding of images”, *IEEE Trans. Inform. Theory,* vol. IT-20, pp. 525-536, July 1974.

[24] N. B. Nill, “A visual model weighted cosine transform for image compression and quality assessment”, *IEEE Trans. Commun.,* vol. COM-33, pp. 551-557, June 1985.

[25] K. N. Ngan, K. S. Leong, and H. Singh, “Cosine transform coding incorporating human visual system model”, presented at SPIE Fiber ’86, Cambridge, MA, pp. 165-171, September 1986.

[26] D. H. Kelly, “Motion and vision II. stabilized spatial-temporal surface”, *J. Opts. Soc. Amer.*, vol. 69, pp. 1340-1349, October 1979.

[27] I. Vujovic, I. Kuzmanic, and M. Krcum, “Experimental Results in Visibility Threshold in Human Visual Perception for Application in Image/Video Coding Quality Assessment”, IEEE Region 8 International Symposium on Video/Image

Processing and Multimedia Communications 16-19, June 2002.

[28] C. –H. Chou and C. –W. Chen, "A Perceptually Optimized 3-D Subband Codec for Video Communication over Wireless Channels", *IEEE transactions on circuits and systems for video technology,* vol.6, no.2, April, 1996.

[29] C. –H. Chou and Y. –C. Li, "A Perceptually Tuned Subband Image Coder Based on the Measure of Just-Noticeable-Distortion Profile", *IEEE Transactions on Circuits and Systems for Video Technology*, vol.5, no.6, December 1995.

[30] A. B. Watson, G. Y. Yang, J. A. Solomon, and J. Villasenor, "Visibility of wavelet quantization noise", *IEEE Transactions on Image Processing,* vol. 6, no. 8, pp. 1164-1175, August 1997.

[31] A. P. Bradley, "A wavelet visible difference predictor", *IEEE Transactions on Image Processing,* vol. 8, no. 5, May 1999.

[32] S. Daly, "The visible difference predictor: An algorithm for the assessment of image fidelity", in *Digital Images and Human Vision*, A. B. Watson, Ed. Cambridge, MA: MIT Press, 1993, pp 176-206.

[33] J. M. Foley and Y. Yang, "Forward pattern masking: Effects of spatial frequency and contrast", *J. Opt. Soc. Amer*. A, vol. 8, no. 12, pp. 2026-2037, December 1991.

[34] M. Kutter and S. Winkler, "A vision-based masking model of spread-spectrum image watermarking", *IEEE Trans. Image Processing,* vol. 11, no. 1, January 2002.

[35] M. Antonini et al, "Image coding using wavelet transform", *IEEE Transactions on Image Processing,* vol. 1, no. 2, pp. 205-221, April 1992.

[36] I. Hontsch and L. J. Karam, "Adaptive image coding with perceptual distortion control", *IEEE Transactions on Image Processing,* vol. 11, no. 3, pp. 213-222,

March 2002.

[37] Z. Liu, L. J. Karam, and A. B. Watsom, “JPEG2000 encoding with perceptual distortion control”, *IEEE Transactions on Image Processing,* vol.1, 14-17, pp. I-637-40, September 2003.

[38] E. Peli, “In search of a contrast metric: Matching the perceived contrast of Gabor patches at different phases and bandwidths,” *Vision Re*s. 37(23), pp. 3217-3224, 1997.

# 作者簡歷

洪朝雄，男，臺灣彰化人，民國七十年一月二十一日生於桃園，家裡共有父母兄弟四人。民國九十二年六月國立交通大學電子工程學系畢業，民國九十二年九月進入國立交通大學電子研究所，從事影像壓縮方面的研究。