RC 21842 (98309) 28 September 2000 ComputerScience

IBM Research Report

Human Visual System Based Rate Scalable Video Coding

Zhou Wang, Larry Lu
IBM T. J. Watson Research Center
P. O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - T. J. Watson - Tokyo - Zurich

IBM Research Report:

Human Visual System Based Rate Scalable Video Coding

Zhou Wang *Larry Lu*

Department of Multimedia Technologies

IBM T. J. Watson Research Center

Abstract

Recently, there are two important trends in video coding research. One is to develop rate scalable video codecs, which allow extraction of coded video data at varying rates from a single compressed stream. The other is the combination of human visual system (HVS) models into the video coder to achieve high visual quality. The motivation behind it is that the widely used measurement, peak signal-to-noise ratio (PSNR), does not correlate well with the perceived video quality.

We are working on a new wavelet based video codec, which takes into account both HVS factors and rate scalability. We propose a framework for HVS-based scalable coding, a foveation model to weight and encode the wavelet coefficients, the method to automatically determine foveation points, a new frame prediction scheme for scalable coding, and a modified SPIHT algorithm. Our new codec has many potential applications and is especially suited for Internet video services, where the video servers must deal with multiple users with variable and time-varying bandwidth requirements.

I. INTRODUCTION

The Internet is a huge and heterogeneous network. Numerous users connect to the Internet by many different ways with very different bandwidth, ranging from as low as 10 Kbps to more than 100 Mbps. In order to provide video services over the Internet (Video over IP), the video server must have the ability to create different bandwidth video stream for different user requirements. One solution to this problem is to compress a video sequence multiple times at different data rates and store multiple versions of the same sequence. The server will then deliver the requested video at the proper rate given the specific available bandwidth. There are several problems with this solution.

1. It requires more resources to be sued on the server in terms of storage space and data management.

2. It does not have the flexibility to adapt to the always-changing network conditions and user requirements.

Therefore, it becomes important to design rate scalable video coding algorithm that allows extraction of coded video data at varying rates from a single compressed stream. With rate scalable coded video stream, the video server can tightly couple the available bandwidth and the data rate of the video being delivered.

In short, rate scalable video coding is the best solution for Internet video communications, because Internet is a *best-effort* network and rate scalable video coding is a *best-effort* technique.

Wavelet based image coding algorithms have achieved great success in recent years. The success relies on the energy compaction feature of the wavelet transforms and the efficient organization, quantization, and encoding of the wavelet coefficients. A typical pyramid wavelet transform structure is given in Fig. 1. Fig. 2 shows its energy compaction feature. A class of embedded coding algorithms has recently received great attention. The most well known algorithms are Shapiro’s embedded zero tree (EZW) algorithm [1], and Said and Pearlman’s set partitioning in hierarchical trees (SPIHT) algorithm [2], which is a refined implementation of the

EZW idea. Embedded wavelet image coding algorithms not only provide very good coding performance, but also have the property of rate scalability. The zero-tree or the hierarchical tree structures in EZW and SPIHT shown in Fig. 3 provide us with a very powerful tool for organizing, ordering and encoding the wavelet coefficients. There are many new wavelet image coding algorithms developed upon EZW or SPIHT [3-7].

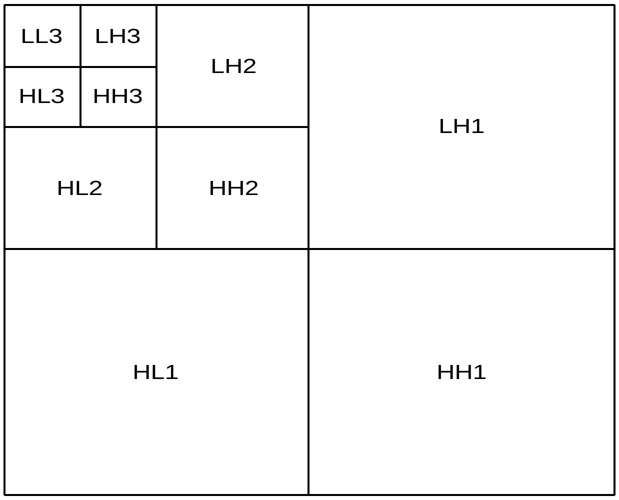


Fig. 1 Pyramid wavelet transform

Fig. 2 Wavelet transform

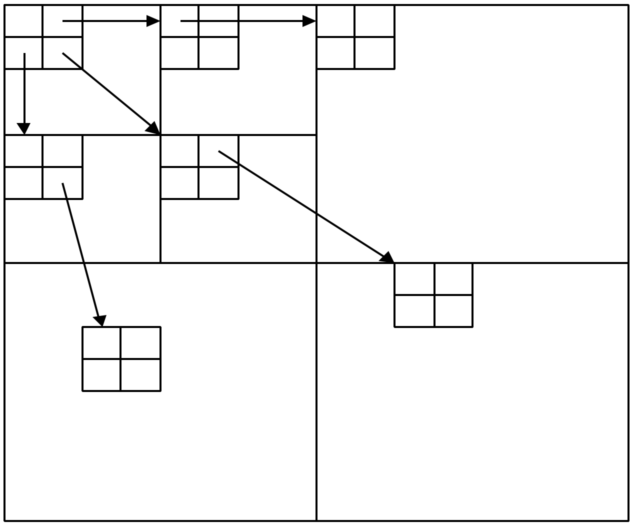


Fig. 3 Zerotree or hierarchical tree structure

Most previous video coding algorithms use the peak signal-to-noise ratio (PSNR) to measure the compressed video quality. However, PSNR and its corresponding distortion measurement of mean squared error (MSE) do not correlate well with the perceived video quality/distortion [8]. Many human visual system (HVS) features must be considered to achieve a good visual quality measurement [9-11]. Accordingly, a good image and video coding algorithm must be designed to optimize the perceived video quality. There are many HVS features that can be used to improve the perceptual video coding algorithms.

First, the error sensitivity of human eyes is a function of spatial frequency. Basically, the function can be viewed as a band pass filter with the frequency response reaching the highest value at about 4 cycles per degree of visual angle and decreasing very fast with increasing spatial frequency. The wavelet transform decomposes the image into frequency subbands. Therefore, the visual sensitivity in different subbands varies.

Second, the HVS's sensitivity to the variations in luminance depends on the local mean luminance. This feature is called light adaptation or luminance masking.

Third, consider two stimuli in the same image. The presence and the features of one stimulus will influence the way the other one is perceived. This is what we called texture masking effect, which is determined by local frequency distribution and texture directions.

Forth, the HVS is highly space-variant in sampling, coding, processing and understanding visual information. The spatial resolution of the HVS is the highest around the point of fixation and decreases rapidly with increasing eccentricity. We call this the foveation feature.

Fifth, the human eyes are more likely to focus on some points of interest such as human faces and moving objects.

Furthermore, there exist temporal HVS features such as the temporal frequency sensitivity, temporal masking effect and the short-term memory effect.

II. HVS-BASED SCALABLE VIDEO CODING ALGORITHM

A. General Framework

The general framework of our encoding algorithm is shown in Fig. 4. First, we divide the whole video sequence into group of pictures (GOP). Each GOP has one intra coding frame (I frame) and the rest are prediction coding frames (P frames).

For encoding I frame, we consider the picture as a still image. We first apply the discrete wavelet transform (DWT) and obtain the wavelet coefficients. The HVS model is employed to determine the visually important points in the image and the importance values of the wavelet coefficients. These values are converted to importance weights to weight the wavelet coefficients. An embedded encoding algorithm (modified SPIHT) is then used to generate the scalable bitstream.

For the P frames, we need to do motion estimation from their previous frames. The result of the motion estimation algorithm is a set of motion vectors. The motion vectors are used to do motion compensation. The motion compensation is done on two versions of the previous frames. One is the original previous frame. The other is a decoded version of the previous frame. The final prediction frame is the weighted combination of the two motion compensation results. The weighting values are given by the HVS model. This is a novel P frame prediction technique and will be discussed in great details later. The wavelet transform is applied to the prediction error frame, and the resulting coefficients are HVS-weighted and coded with the modified SPIHT algorithm.

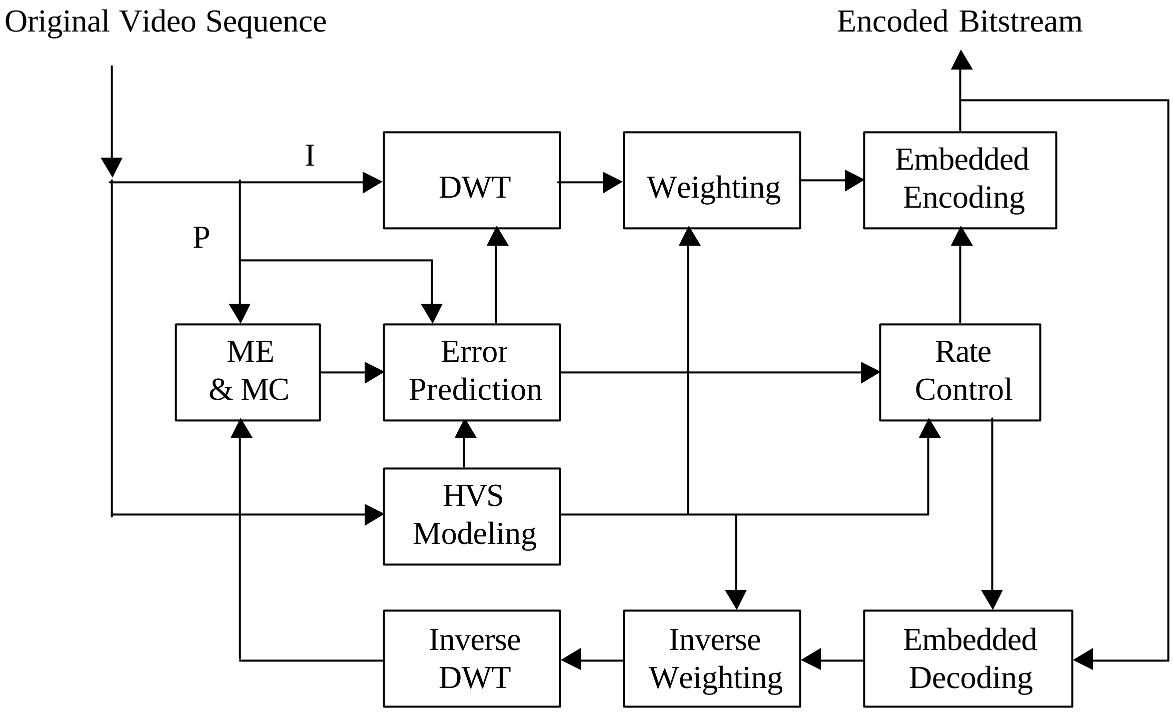


Fig. 4 General framework of the encoding algorithm

During the encoding process, a rate control algorithm is used to allocate bits to each frame. The allocation is determined by the available bandwidth, the HVS modeling results and the frame prediction error.

B. Automatic Determination of Important Regions

Every point in the picture could be visually very important. However, it is not practical to exam all the points. One reason is the high computational complexity. The other reason is that we have to use too many bits to encode the coordinates of all the selected points. We choose centers of all the macro blocks as candidate foveation points to limit the computation and we can use only one bit for each macro block to encode this information.

The methods to find foveation points for I frames and P frames are different. For I frames, we first detect the points that is interesting to human eyes. In the current work, human faces are assumed to be of interest. Also, bright areas are selected as important regions because human eyes are more sensitive to the errors occurred in bright areas.

For P frames, the most important information is how much new information is given in comparison with the prediction frame. Therefore, bad motion prediction regions are selected as foveation points. We also care more about the regions of interest, but if the motion prediction from the previous frame is already good enough, it is not necessary to encode the same information multiple times. Therefore, we only put foveation points for those points on the areas with certain amount of prediction errors. Different from other regions, only small prediction error may result in a foveation point within the regions of interest.

An example of the foveation point detection results for an original I frame (Fig. 5) is given in Fig. 6, and another example for a P frame is shown in Fig. 7.

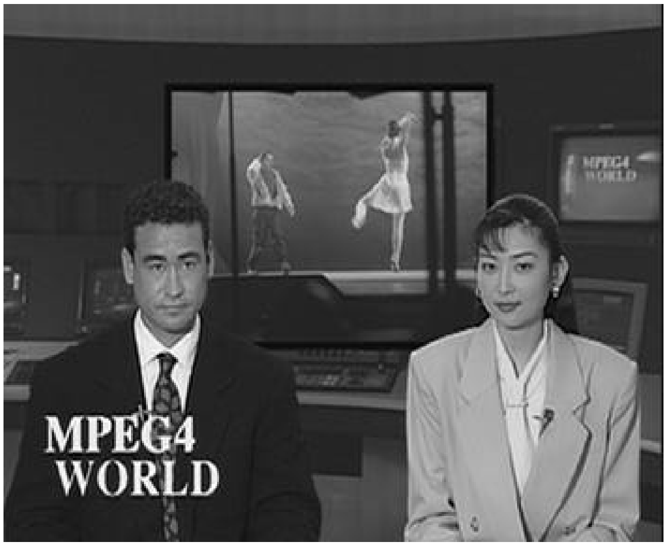


Fig. 5 A sample image frame in the sequence “News”.

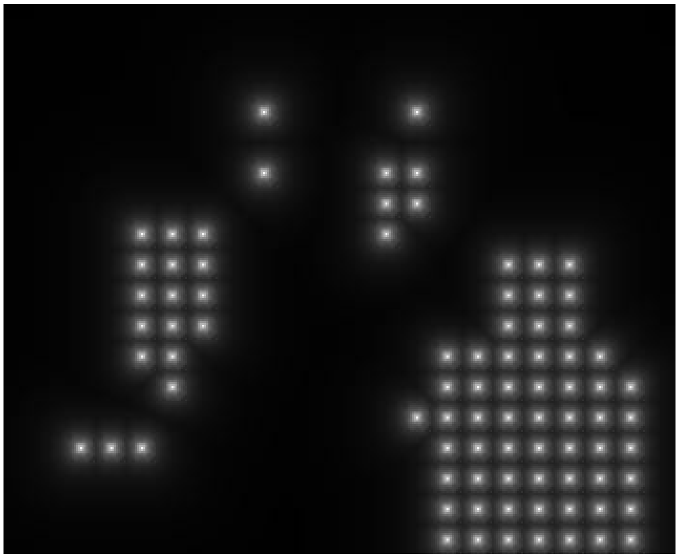


Fig. 6 I frame foveation point detection

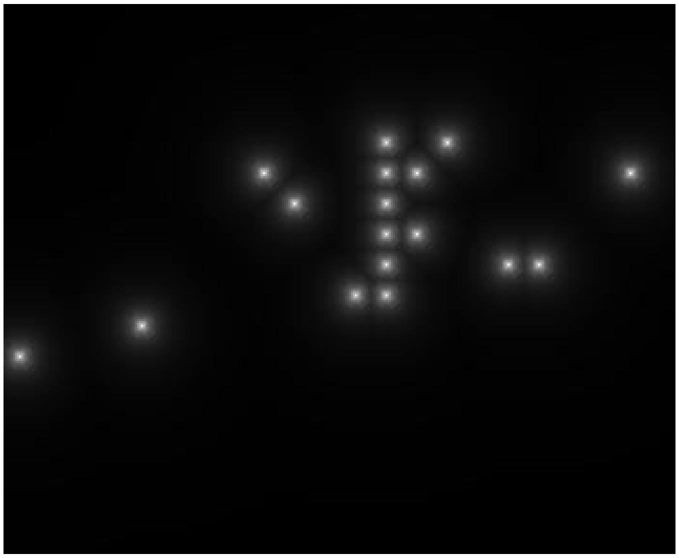


Fig. 7 P frame foveation point detection

C. Face Detection

Face detection is an important part in the determination of the foveation points. It can significantly improve the visual quality for some special environment such as video conferencing.

Our face detection algorithm is composed of three steps. The first step is to identify the possible face regions by the skin color information [12]. The result of the first step is actually a binary image, with each pixel indicating itself as skin-color pixel or not. In the second step, we detect human face in those skin-color regions by a technique called binary template matching [13]. Finally in the last step, we verify every detected face and remove the falsely detected faces. The verification is based on the observation that human face areas usually have certain amount of high frequency component distributions because of the existence of discontinuities at the eyes, the nose, and the mouth.

D. Foveation Resolution Model

The photoreceptors (cones and rods) and ganglion cells are non-uniformly distributed in the retina in the human eye. The density of cones and ganglion cells [9, 10] play key roles in determining the ability of our eyes to resolve what we see. Spatially, the resolution, or sampling density, has the highest value at the point of the fovea and decreases rapidly away from that point as a function of eccentricity. As a result, when a human observer gazes at a point in a real world image, a variable resolution image is transmitted through the front visual channel into the high level processing units in the human brain. The region around the point of fixation (or foveation point) is projected into the fovea and sampled with the highest density. In conclusion, the HVS is space-variant in sampling, coding, processing and understanding visual information. By contrast, traditional digital computer vision systems represent images on rectangular uniformly sampled lattices, which have the advantages of simple acquisition, storage, indexing and computation. Nowadays, most digital images and videos are stored, processed, transmitted and displayed in rectangular matrix format, where each entry represents one sampling point.

So far, people have been doing valuable research work on foveation image and video processing [14-19, 24-30]. One method is foveation filtering, which aims to foveate a uniform resolution image, such that when the human eyes gaze at the point of fixation, they cannot distinguish between the original and the foveated versions of that image. Another research focus is foveation image and video compression. In fact, some of the foveation filtering processes, themselves reduce a large amount of information. Even higher compression can be obtained when we combine them with other signal compression techniques. In [27], a fast video coding system is proposed that uses Laplacian pyramid coding following foveation. In [17-19, 24-26], MPEG/H.263 video coding is applied to foveation filtered video sequences.

In order to develop a foveation resolution model, we need to consider the anatomy of the early vision system. The light first passes through the optics of the eye and is then sampled by the photoreceptors on the retina. There are two kinds of photoreceptors – cones and rods. The cone receptors are responsible for the daylight vision. Their distribution is highly non-uniform on the retina. The density of the cone cells is the highest at the fovea and drops very fast with increasing eccentricity. The photoreceptors deliver data to the bipolar cells, which in turn supply information to the ganglion cells. The distribution of ganglion cells is also highly non-uniform. In the fovea, approximately two to three ganglion cells receive their inputs from the same photoreceptor. The density of the ganglion cells drops even faster than the density of the cone receptors. The variation of the densities of photoreceptors and ganglion cells with eccentricity is shown in Fig. 8. The human eyes' resolution ability is determined by these density distributions.

The goal of our foveation resolution model is to calculate the local resolution for each point in an observed image. For simplicity, we assume the observed image is N pixels in width and the line from the fovea to the point of fixation in the image is perpendicular to the image plane. To calculate the eccentricity of a given point $\boldsymbol{x} = (x_1, x_2)^T$, where x_1 and x_2 are measured in pixel width, we assume we know the position of the foveation point $\boldsymbol{x}^{\boldsymbol{f}} = (x_1^f, x_2^f)^T$ (pixels) in the image and the viewing distance d_v (measured in image width) from the eye to the image plane. The distance d_x (in image width) from $\boldsymbol{x}$ to $\boldsymbol{x}_f$ can then be computed as $d_x = \frac{r(\boldsymbol{x})}{N}$, where $r(\boldsymbol{x}) = \left\|\boldsymbol{x} - \boldsymbol{x}^{\boldsymbol{f}}\right\|_2 = \sqrt{(x_1 - x_1^f)^2 + (x_2 - x_2^f)^2}$ (in pixels). The eccentricity is

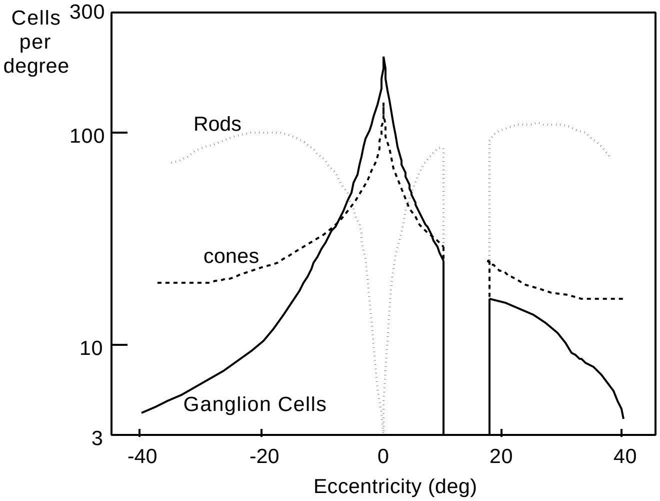


Fig. 8 Photoreceptor and ganglion cell distribution in human retina.

$$e_x = \tan^{-1}\left(\frac{d_x}{d_v}\right) = \tan^{-1}\left(\frac{r(\boldsymbol{x})}{Nd_v}\right). \tag{1}$$

The local resolution can be represented by the maximum local detectable frequency or by the maximum local sampling density without aliasing. They are equivalent because the former is one half of the latter due to the sampling theorem. There have been many efforts to measure and model local resolution experimentally as a function of eccentricity [14-18]. We use the following function to model the maximum local detection frequency for a given pixel ***x***, which is simplified and normalized from the model in [14]:

$$f_m(\boldsymbol{x}) = f_m[r(\boldsymbol{x})] = \frac{1}{1 + C_1 \cdot e_x}, \tag{2}$$

where C_1 is a constant. The resolution value is normalized, so that the highest value is equal to 1 at zero eccentricity. We normalize it because only the relative resolution is of interest in our algorithms. Fig. 9 gives f_m as a function of r for fixed N = 512 and d_v = 1, 2, and 4, respectively.

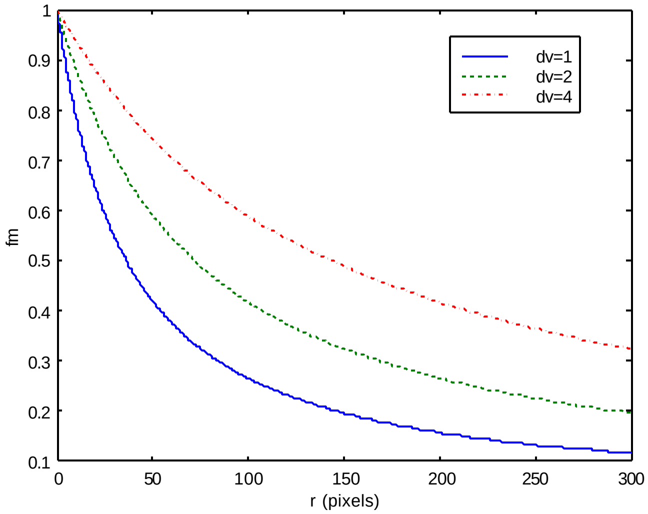


Fig. 9 Foveation resolution model

In addition, the maximum perceived resolution of an observed image is also limited by the display resolution f_d , which is also a function of viewing distance. It is calculated as:

$$f_d = C_2 N \Big/ \tan^{-1}\left(\frac{1}{2d_v}\right). \tag{3}$$

Finally, the maximum local frequency for a given pixel $\boldsymbol{x}$ in the image is given by:

$$f(\boldsymbol{x}) = f[r(\boldsymbol{x})] = \min(f_m(r(\boldsymbol{x})), f_d) \tag{4}$$

E. HVS-Based Weighting Model in Wavelet Transform Domain

E.1 HVS, wavelet transform and image coding

The discrete wavelet transform (DWT) has proved to be a powerful tool for image compression. In DWT-based image coding, a multilevel 2-D wavelet transform is first used to decompose the original image into subbands. Recently, the 9/7 biorthogonal filters [20] have been widely adopted for DWT-based image compression algorithms. We also use the 9/7 filters in this paper. The outputs of the wavelet transform (called wavelet coefficients) are reorganized, quantized, and entropy coded. EZW and SPIHT are good examples for the encoding of wavelet coefficients.

The wavelet coefficients at different subbands and locations supply the information of variable perceptual importance to the HVS. In order to develop a good wavelet-based image coding algorithm that considers HVS features, we need to measure the visual importance of the wavelet coefficients. The more important coefficients are assigned larger weights, and the less important coefficients are assigned smaller weights.

E.2 HVS-based weighing model for intra-coding frames

Human visual sensitivity to errors is a function of spatial frequency. The wavelet transform decomposes the image into frequency subbands. Therefore, the visual sensitivity in different subbands varies. In [21], psychovisual experiments were conducted to measure the visual sensitivity in wavelet decompositions. Noise was added to the wavelet coefficients of a blank image with uniform mid-gray level. After the inverse wavelet transform, the noise threshold in the spatial domain is tested. A model that provides a reasonable fit to the experimental data is [21]:

$$\log Y = \log a + k(\log f - \log g_q f_0)^2 \tag{5}$$

where Y is the visually detectable noise threshold, q is an orientation index, representing LL, LH, HH, and HL subbands, respectively, and f is the spatial frequency measured in cycles/degree. It is determined by the display resolution r and the wavelet decomposition level l : $f = r2^{-l}$ [21]. The parameters are tuned to fit experimental data. For gray scale models, a is 0.495, k is 0.466, f_0 is 0.401, and g_q is 1.501, 1, and 0.534 for the LL, LH/HL, and HH subbands, respectively. The error detection thresholds for the wavelet coefficients can be calculated by:

$$T_{l,q} = \frac{Y_{l,q}}{A_{l,q}} = \frac{a 10^{k(\log(2^l f_0 g_q / r)^2}}{A_{l,q}} \tag{6}$$

where $A_{l,q}$ is the basis function amplitude given in [21]. It is typical to define the error sensitivity as the inverse of the error detection threshold. Therefore, we define the error sensitivity in subband (l,q) as:

$$S_{l,q} = \frac{1}{T_{l,q}} \tag{7}$$

Within each wavelet subband, we determine the importance of each coefficient using the foveation resolution model. The first step is to find the corresponding foveation point in each subband. For the pyramid wavelet decomposition structure, we have:

$$\text{LL:} \quad \boldsymbol{x}_{l,q}^{f} = \left(\frac{x_1^f}{2^l}, \frac{x_1^f}{2^l} \right)^T; \qquad \text{LH:} \quad \boldsymbol{x}_{l,q}^{f} = \left(\frac{x_1^f + N}{2^l}, \frac{x_1^f}{2^l} \right)^T;$$

$$\text{HL:} \quad \boldsymbol{x}_{l,q}^{f} = \left(\frac{x_1^f}{2^l}, \frac{x_1^f + N}{2^l} \right)^T; \qquad \text{HH:} \quad \boldsymbol{x}_{l,q}^{f} = \left(\frac{x_1^f + N}{2^l}, \frac{x_1^f + N}{2^l} \right)^T.$$

Next, for a given wavelet coefficient at position $\boldsymbol{x}$, which resides in subband (l,q), its equivalent distance from the foveation point in the spatial domain is $r_{l,q}(\boldsymbol{x}) = 2^l \left\| \boldsymbol{x} - \boldsymbol{x}_{l,q}^{f} \right\|_2$. For this equivalent distance, we define the importance according to the foveation resolution model given by (4):

$$Q(\boldsymbol{x}) = f(r_{l,q}(\boldsymbol{x})). \tag{8}$$

Finally, the importance weight for the wavelet coefficient at $\boldsymbol{x}$ is determined by a combined consideration of the error sensitivity model for different subbands and the foveation resolution model:

$$W(\boldsymbol{x}) = [S_{l,q}]^a [Q(\boldsymbol{x})]^b, \tag{9}$$

where a and b are parameters used to control the magnitudes of $S_{l,q}$ and $Q(\boldsymbol{x})$, respectively. The construction of the weighting model can be viewed as two stages in cascade. In the first stage, each wavelet subband is assigned a uniform base importance weight according to the error sensitivity model for the subbands. In the second stage, non-uniform weights developed from the foveation resolution model are applied to the subbands, resulting in a space-variant importance weighting mask in the wavelet transform domain.

In Fig. 10 and Fig. 11, we show examples of the resulting importance weighting models for I frame and P frame, respectively.

F. Modified SPIHT

We use a modified SPIHT algorithm to do the embedded encoding. The reason to choose the modified SPIHT instead of the original SPIHT is that the original SPIHT becomes less efficient after the HVS-based weighting. We need to go through the details of EZW and SPIHT in order to illustrate our modified SPIHT algorithm.

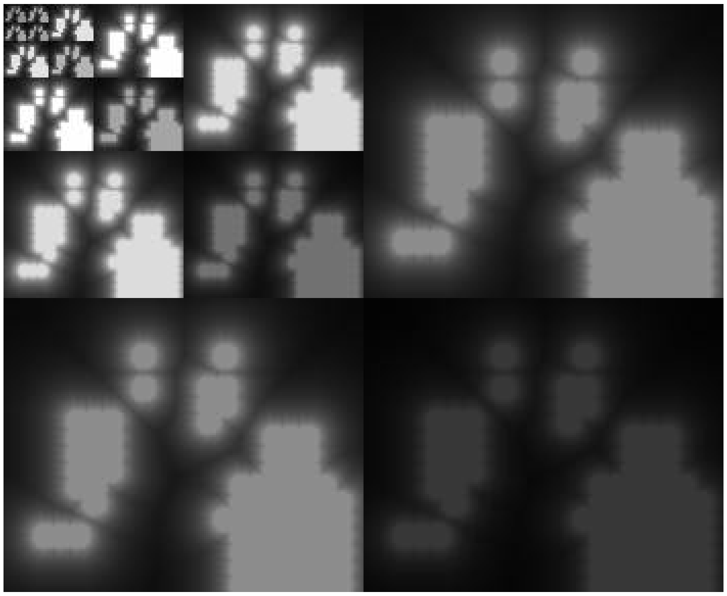


Fig. 10 I frame HVS weighting in wavelet transform domain

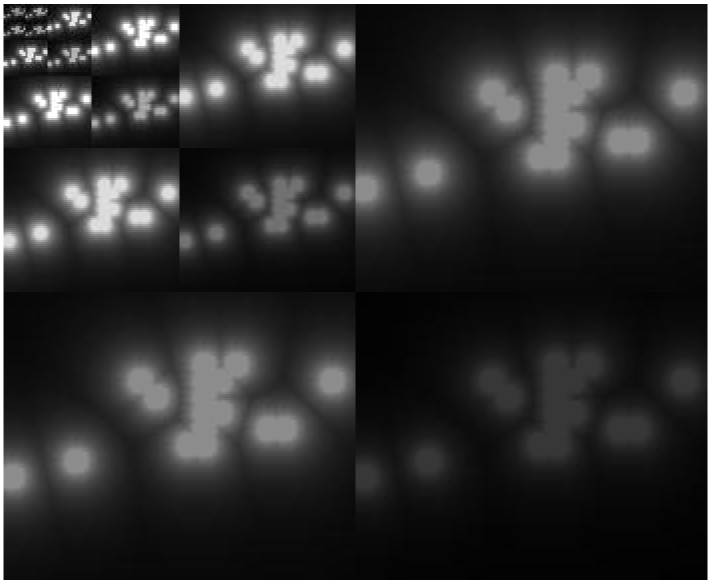


Fig. 11 P frame HVS weighting in wavelet transform domain

F.1 Embedded wavelet image coding

With the embedded encoding algorithm, the encoded bitstream is ordered such that the most important bit to the HVS comes first, followed by the second important one, and so on. Of course, the overhead information such as the video sequence header, frame headers, possibly group of frame headers, and some HVS modeling parameters is still necessary. This kind of overhead information is not scalable and should be encoded using only a very small number of bits.

The main objective in embedded wavelet image coding is to choose the most important wavelet coefficients to be encoded and transmitted first. The importance of a coefficient in EZW and SPIHT depends on its contribution to the MSE distortion. It is shown in [22] that the coefficients with larger magnitudes are more important. The strategy is ordering the coefficients by magnitude and transmitting the most significant bits first. Assume that the wavelet coefficients have been ordered according to the minimum number of bits required for its magnitude binary representation. The most effective order for progressive transmission is to sequentially send the bits in each row. In order for the decoder to understand the meaning of the bits, we also need to encode and transmit the coordinates of the wavelet coefficients along with the magnitude bits. It has been observed that the wavelet coefficients that are less significant have structural similarity across the wavelet subbands in the same spatial orientation. The zero tree structure in EZW and the spatial orientation tree structure in SPIHT capture this structural similarity very effectively. Fig. 3 is the spatial orientation tree used by SPIHT.

In EZW or SPIHT encoder, the wavelet coefficients are scanned multiple times. Each time consists of a sorting pass and a refinement pass. The sorting pass selects the significant coefficients and encodes the spatial orientation tree structure. A coefficient is significant if its magnitude is larger than a threshold value, which decreases by a factor of 2 for each successive sorting pass. The refinement pass outputs one bit for each selected coefficient. An entropy coder can be used to further compress the output bitstream. SPIHT performs better than EZW in terms of reconstructed image quality. The coding part of our algorithm is developed based on a modified version of SPIHT.

F.2 Modified SPIHT algorithm

It is possible to implement an embedded foveation coding system with the original SPIHT algorithm. However, there arises a problem that makes the SPIHT algorithm inefficient. Note that in our system, the input coefficients to the SPIHT part are the weighted wavelet coefficients instead of the original wavelet coefficients. The weighted coefficients have a much larger dynamic range in comparison to that of the original coefficients. The change of dynamic range leads to an increase in the number of times of scanning the wavelet coefficients. This makes SPIHT encoding less efficient in two aspects. First, since we need to encode the spatial orientation tree structure every scanning time, an increase in the number of times an image is scanned implies an increase in the wastage of bits and hence an increase of time for scanning and computation. Second, we are encoding the significant coefficients with more bits as we increase the number of scanning because we add one more refinement bit to each of them during each scanning time. Consequently, a modified SPIHT algorithm is needed to overcome this problem. We solve this problem in two ways.

First, in the sorting pass, we do not scan all the wavelet coefficients in the first few scanning times. Suppose the maximum absolute value of the unweighted wavelet coefficients is $\max_{\boldsymbol{x}}\{c(\boldsymbol{x})\}$, then the largest possible absolute value of weighted wavelet coefficient at location $\boldsymbol{x}$ is $\max_{\boldsymbol{x}}\{c(\boldsymbol{x})\} \cdot W(\boldsymbol{x})$. In the first few scanning times, it might be lower than the significance test threshold in the SPIHT algorithm. The threshold for the n-th time of scanning is given by $T_n = \lfloor \log_2 (\max_{\boldsymbol{x}}\{c(\boldsymbol{x})W(\boldsymbol{x})\}) \rfloor / 2^{n-1}$. Our strategy is that before each time of scanning, we perform the following test: $\max_{\boldsymbol{x}}\{c(\boldsymbol{x})\} \cdot W(\boldsymbol{x}) \geq T_n$? Only those $\boldsymbol{x}$ satisfy this condition are scanned. The wavelet coefficients that are scanned the first four times are given in Fig. 12. It can be seen that only a small subset of the coefficients needs to be scanned during the first few sorting passes.

Second, in the refinement pass, it is not necessary to encode any of the weighted coefficients using many bits. This is because during the inverse weighting procedure, the coefficients will be scaled to values on the order of their original values. The less significant bits (such as the 10-th or even less significant bits) do not have much contribution to the overall image quality, therefore can be removed. In our algorithm, we limit the maximum number of bits for each coefficient. Actually, any wavelet coefficient that receives the upper limit of refinement bits can be removed from the list of significant pixels of the SPIHT algorithm.

G. Motion Estimation

The motion estimation part of our algorithm is done in two stages. First we do 16 by 16 macro block full search motion estimation. After finding the best match block, we calculate its prediction error. If the error is bigger than a threshold, we split the macro block into four 8 by 8 blocks and estimate the motion vectors for the four blocks, respectively.

In order to encode the motion vectors, we first map the 2-D coordinate of each motion vector into a single value, so that we can represent the motion vector with a 1-D indexing value. We then use an adaptive arithmetic coding algorithm to compress the 1-D indexing values.

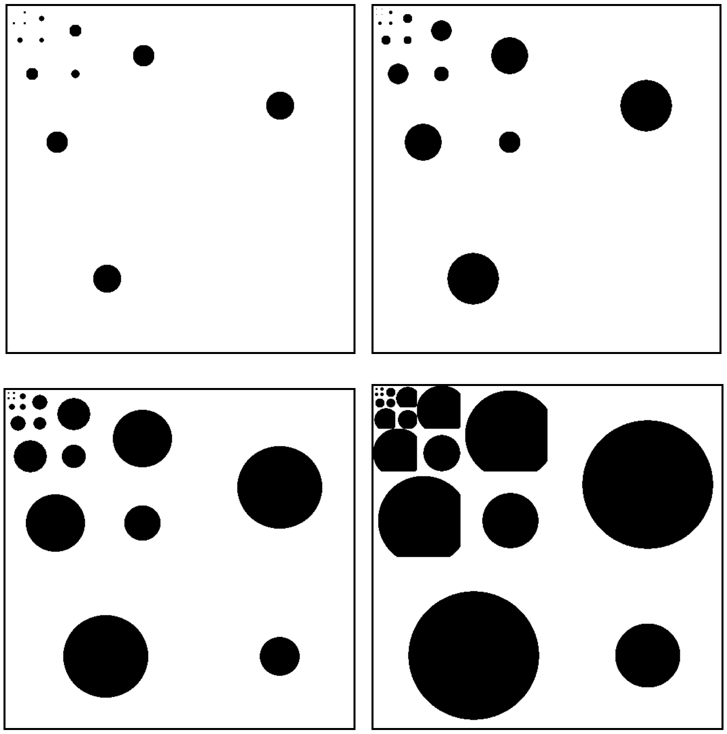


Fig. 12 The wavelet coefficients scanned in the first four scanning time of the modified SPIHT algorithm.

H. New Frame Prediction Algorithm

Frame prediction plays a key role in motion estimation/motion compensation based video coding. In rate scalable video coding, frame prediction becomes a more complicated problem in comparison with fixed-rate video coding. The reason is that at the encoder side, we do not have any information about the decoding bit rate at the decoder side.

There are several solutions to this problem.

In the first solution, we can use the original motion compensated frames as the prediction. This method is very simple to implement. However, the prediction frames at the encoder and the decoder sides are different, sometimes very different, especially at the difficult encoding regions. Therefore, the prediction is not good. An even bigger problem is it causes significant error drifting. That is, the errors generated in this frame will propagate to all the prediction frames followed.

The second solution is to use a lower base bit rate decoded and motion compensated frame as the prediction. The advantage of this solution is that the prediction frames at the encoder and the decoder are exactly the same. Therefore, error drifting problem is avoided. However, if the decoding bit rate is much higher than the base bit rate, then very big prediction errors will occur. Thus the prediction errors are big, and the encoding is not efficient. The idea of solution 2 has been used in [23].

We propose a new solution to this problem. In our algorithm, the original motion compensated frame and the lower base bit rate decoded and motion compensated frame are combined to make a prediction. The combination is based on our HVS modeling. Our new frame prediction algorithms at the encoder and the decoder are shown in Fig. 13 and Fig. 14, respectively.

At the encoder, we have two reference frames. One is the previous frame from the original sequence, and the other is the previous frame decoded from the base bit rate. The same motion compensation process is applied to both of them and generate too motion compensated reference frames. These two frames are combined by the HVS weighting model. The weighting idea is that for the easy prediction regions, we give more weights to the originally motion compensated reference frame. Although there exists error drifting problem in these regions, but the error is small, hence the drifting problem is small. For the regions that are difficult to encode, we give more weights to the base bit rate motion compensated reference frame. The errors generated in these regions are usually large, but will not drift to the frames that followed.

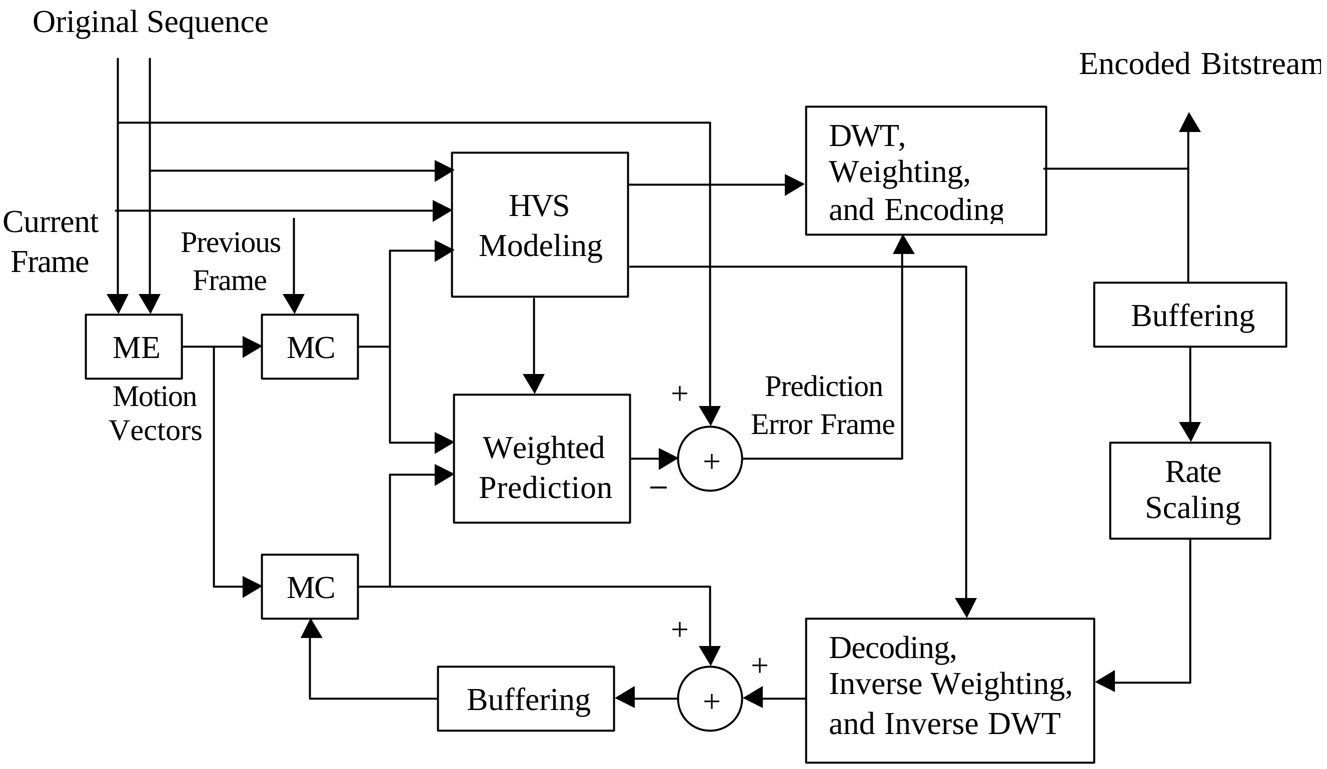


Fig. 13 New frame prediction algorithm - Encoder side.

The decoder also makes use of two reference frames. One is the previous frame decoded from the base bit rate. The other is the previous frame decoded at the current decoding bit rate. The motion vectors are decoded and used by the motion compensation block to generate two versions of the reference frames, one from the base bit rate version, and one from the current decoding bit rate version. The HVS information is also decoded and used to calculate the HVS model, which is exactly the same as the encoder. The motion compensated two reference frames are then weighted with the HVS model and create the prediction frame. The decoded prediction error frame is added to the prediction frame to achieve the current decoded frame.

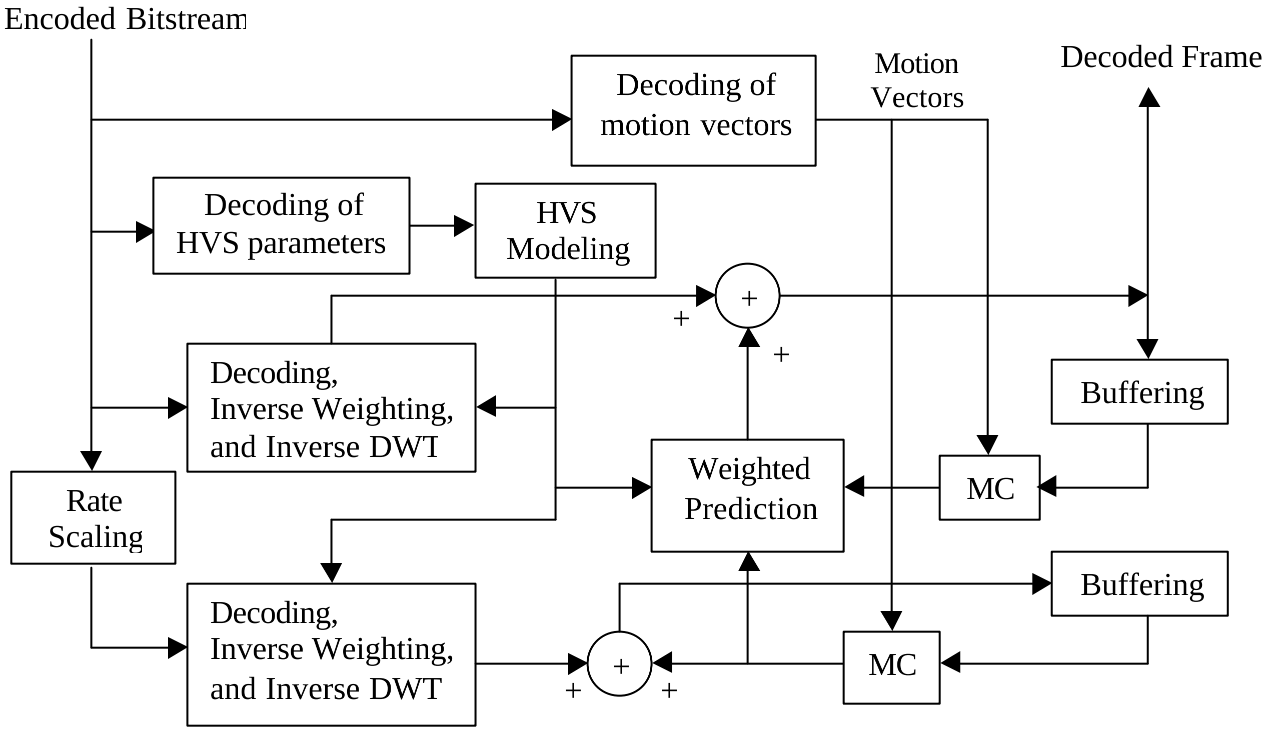
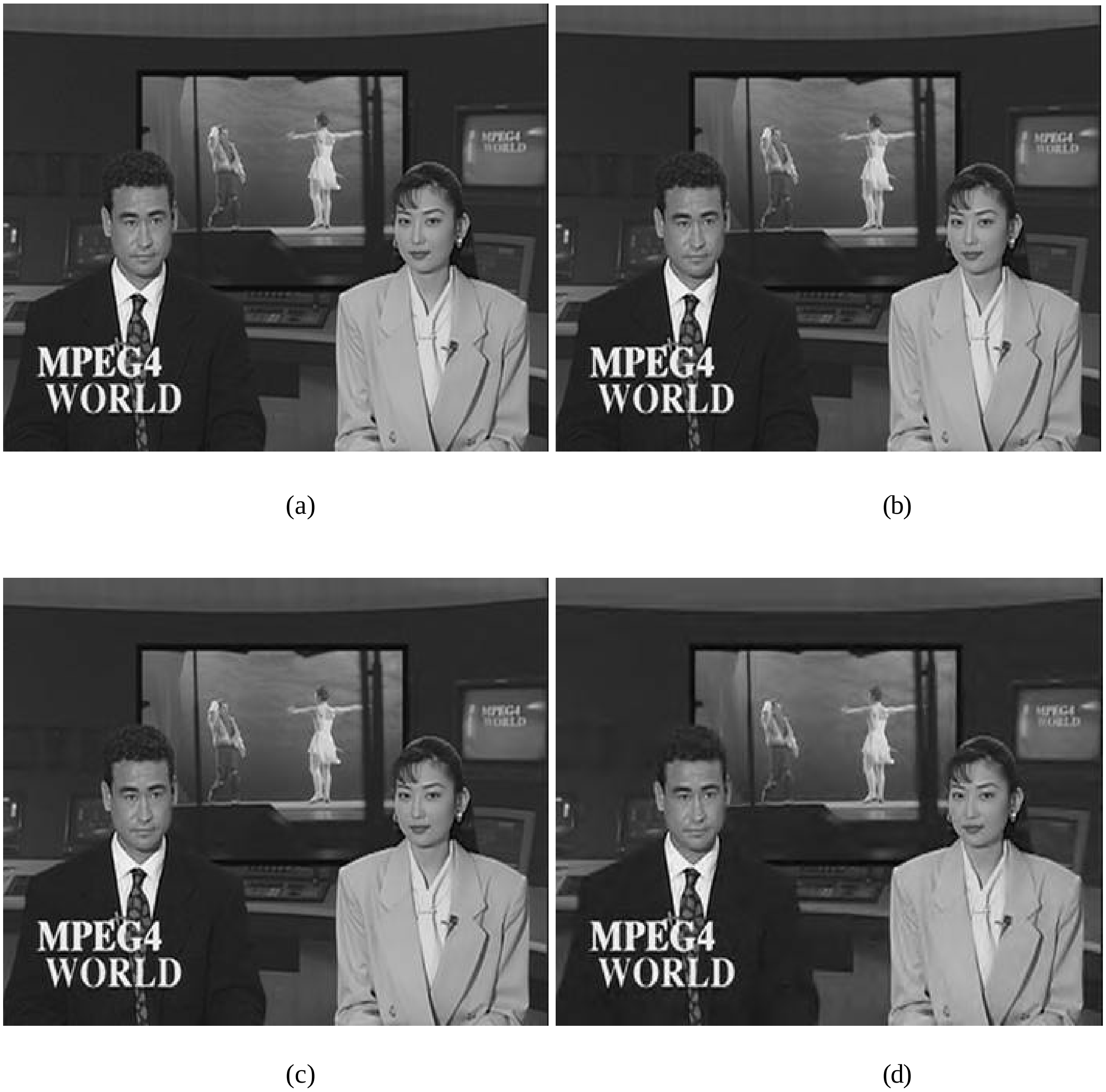

Encoded Bitstream
Decoding of motion vectors
Motion Vectors
Decoded Frame
Decoding of HVS parameters
HVS Modeling
+
+
+
Decoding, Inverse Weighting, and Inverse DWT
Buffering
Rate Scaling
Weighted Prediction
MC
Buffering
Decoding, Inverse Weighting, and Inverse DWT
+
+
+
MC

Fig. 14 New frame prediction algorithm - Decoder side.

Our new algorithm does not avoid error drifting problem completely, but it becomes very small and can be neglected. The good thing is that we have much better prediction, which leads to small prediction error and better compression performance.

I. Results

We tested our algorithm on CIF size (288×352), YCbCr 4:2:0 format video sequences. Although our algorithm is not as sophisticated as the MPEG and H.263 standards, especially with respect to the motion estimation, motion vector encoding and rate control parts, comparable visual quality to MPEG-1 and H.263 is obtained at 500Kbps or higher compression for general video sequences. For special video sequences like the 'News' sequence, enhanced visual quality is obtained because our coder can exploit more HVS features. Fig. 15 shows the decoded luminance frame 18 of the 'News' sequences at 600Kbps, 400Kbps and 200Kbps, respectively, where the base bit rate is 200Kbps and the original sequence is encoded to 2Mbps and then scaled down to multiple versions with different bit rates. However, our method has many potential improvements, especially at the motion estimation and motion vector compression parts, which are very critical at low bit rate coding.



(a) (b)

(c) (d)

Fig. 15 Frame 18 of the coded ‘News’ sequence. Encoded at 2Mbps with 200Kbps base bit rate. (a) original frame; (b) decoded at 600Kbps; (c) decoded at 400Kbps; (d) decoded at 200Kbps.

III. FUTURE WORK

There are a lot potential improvement we can do based on our current encoder.

Firstly, we can improve the motion estimation part. Currently, we only use integer pixel motion estimation. In many cases, especially the fast and abundant motion cases, half pixel motion estimation can significantly reduce the prediction error. Our current algorithm does not have intra-coding mode. This is not good for the cases where prediction errors are very big or scene cuts occur. Therefore, we can add half pixel and intra-coding to have smaller prediction errors and hence improve compression efficiency.

Secondly, our current method to encode motion vectors is not efficient. We have many ways to improve it. For example, we can encode the difference of adjacent motion vectors instead of encoding them directly. Also if we have good statistics estimates, we can design a Huffman table to encode the difference of the motion vectors.

Thirdly, bi-directional frame (B frame) predication is a very important part of the MPEG coding standard. We believe it is also very helpful for our algorithm.

Fourthly, many other HVS features, such as texture masking effect and temporal masking effect, can be added to improve visual quality.

Finally, we currently use a very simple rate control method to allocate bits to each frame. A more sophisticated rate control algorithm is useful for optimal bit allocation.

With all or several of these improvements, we expect to have a robust video coding algorithm that has comparable visual quality to the existing standards for general videos and better visual quality over current standards for specific video communication environments such as videoconferencing.

With our new video codec, there can also be many potential new applications.

One new application is that we can do knowledge based video coding. That means the more we have the knowledge about the video sequences to be encoded, the more we can do to improve the visual quality because the knowledge can be very simply and naturally embedded into our basic model of the codec.

Another new application is dynamic variable bit rate video transmission. Since the bandwidth connection over the Internet is not only non-uniform, but also changes with time, it is very preferable to have a video coder that can dynamically change the bit rate with time. Our rate scalable coder is good for this purpose.

In order for our algorithm to be effectively used over the Internet, we need to work on the video transmission protocols on the transport and upper layers of the Ineternet.

We believe our HVS based rate scalable video coding is the best solution for Video Over IP.

IV. SUMMARY

We summarize our research on our new video coding technique as follows:

First, we introduced a new framework for HVS-based rate scalable wavelet video coding.

Second, we proposed an HVS-based model to weight and encode the wavelet coefficients.

Third, we developed some new algorithms to automatically determination of the important points and regions in the video sequences to be compressed. Two different determination strategies are used for I frames and P frames.

Fourth, a novel frame prediction algorithm for rate scalable video coding is proposed.

Furthermore, a modified SPIHT algorithm is developed to improve the embedded coding method of SPIHT.

Besides the above new research work, we also finished many implementation work using existing techniques.

We implemented a very efficient face detection algorithm.

We developed a two-stage motion estimation algorithm.

We wrote the software of the whole encoding and decoding system using Microsoft Visual C++. The source codes have more than 7000 lines.

In conclusion, we have developed the first version of a very promising rate scalable video codec, which has many potential improvement and applications.

References

[1] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Transactions on Signal Processing*, vol. 41, no. 3445-3462, Dec. 1993.

[2] A. Said, and W. A. Pearlman, “A new, fast and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243-250, June 1996.

[3] S. G. Mallat, “Multifrequency channel decomposition of images and wavelet models,” *IEEE Trans. Acoustics., Speech, Signal Processing*, vol. 37, pp. 2091-2110, 1989.

[4] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using the wavelet transform,” *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 205-220, 1992.

[5] J. Li, and S. Lei, “An embedded still image coder with rate-distortion optimization,” *IEEE Trans. Image Processing*, vol. 8, no. 7, pp. 913-924, July 1999.

[6] J. Villasenor, b. Belzer, and J. Liao, “Wavelet filter evaluation for efficient image compression,” *IEEE Trans. Image Processing*, vol. 4, pp. 1053-1060, 1995.

[7] Z. Wang and A. C. Bovik, “Embedded foveation image coding,” submitted to *IEEE Transactions on Image Processing*, Apr. 2000.

[8] B. Girod, “What’s wrong with Mean-Squared Error,” *Digital Images and Human Vision*, Chapter 15, A. B. Watson, Ed., 1993.

[9] B. A. Wandell, *Foundations of Vision*, Sinauer Associates, Inc., 1995.

[10] W. S. Geisler, *Vision Notes*, The University of Texas at Austin, 1999.

[11] Z. Wang and A. C. Bovik, “A human visual system based objective video distortion measurement system,” *International Conference on Multimedia Processing and Systems*, Aug. 2000.

[12] C. Garcia, and G. Tziritas, “Face detection using quantized skin color regions merging and wavelet packet analysis,” *IEEE Transactions on Multimedia*, vol. 1, no. 3, pp. 264-277, Sep. 1999.

[13] H. Wang, and S.-F. Chang, “A highly efficient system for automatic face region detection in MPEG Video,” *IEEE Trans. on Circuits and System for Video Technology*, vol. 7, no. 4, pp. 615-628, 1997.

[14] P. Kortum, W. S. Geisler, “Implementation of a foveated image coding system for image bandwodth reduction,” *Human Vision & Electronic Imaging, Proc. of SPIE*, vol. 2657, pp. 350-360, 1996.

[15] T. Kuyel, W. Geisler, and J. Ghosh, “Retinally reconstructed images: digital images having a resolution match with the human eyes,” *IEEE Trans. System, Man & Cybernetics, - Part A: Systems and Humans*, vol. 29, no. 2, pp.235-243, Mar. 1999.

[16] T. Kuyel, “Foveated models in compression, texture discrimination and classification,” *Ph.D. Dissertation*, The University of Texas at Austin, May 1997.

[17] S. Lee, “Foveated video compression and visual communications over wireless and wireline networks,” *Ph.D. Dissertation*, The University of Texas at Austin, May 2000.

[18] S. Lee, M. S. Pattichis and A. C. Bovik, “Foveated video quality assessment and compression gain”, submitted to *IEEE Transactions on Multimedia*, 2000.

[19] S. Lee, M. S. Pattichis and A. C. Bovik, “Optimal rate control for real-time, low bitrate foveated video,” submitted to *IEEE Transactions on Image Processing*, 2000.

[20] A. Cohen, I. Daubechies, and J. C. Feauveau, “Biorthogonal bases of compactly supported wavelets,” *Commun. Pure Appl. Math.*, vol. 45, pp. 485-560, 1992.

[21] A. B. Watson, G. Y. Yang, J. A. Solmon, and J. Villasenor, “Visibility of wavelet quantization noise,” *IEEE Trans. Image Processing*, vol. 6, no. 8, pp. 1164-1175, Aug. 1997.

[22] A. P. Bradley, "A wavelet visible difference predictor," *IEEE Trans. Image Processing*, vol. 8, no. 5, pp. 717-730, May 1999.

[23] K. S. Shen, and E. J. Delp, "Wavelet based rate scalable video compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 1, pp. 109-122, Feb. 1999.

[24] S. Lee, A. C. Bovik and Y. Y. Kim, "High quality, low delay foveated visual communications over mobile channels," *IEEE Transactions on Multimedia*, in press, 2000.

[25] S. Lee, M. S. Pattichis and A. C. Bovik, "Rate control for foveated MPEG/H.263 video," *IEEE International Conference on Image Processing*, vol. 2, pp. 365-369, 1998.

[26] S. Lee, A. C. Bovik and Y. Y. Kim, "Low delay foveated visual communications over wireless channels," *IEEE International Conference on Image Processing*, Oct. 1999.

[27] W. S. Geisler, and J. S. Perry, "A real-time foveated multiresolution system for low-bandwidth video communication," *Proceedings of SPIE*, vol. 3299, 1998.

[28] E.-C. Chang, and C. Yap, "A wavelet approach to foveating images," *Proceedings of the 13th ACM Symposium on Computational Geometry*, pp.397-399, 1997.

[29] E.-C. Chang, "Foveation techniques and scheduling issues in thinwire visualization," *Ph.D. Dissertation*, New York University, May 1998.

[30] E.-C. Chang, S. Mallat, and C. Yap, "Wavelet foveation," http://www.cs.nyu.edu/visual/, Jan. 1999.