Foveation Embedded DCT Domain Video Transcoding

Shizhong Liu and Alan C. Bovik *

Laboratory for Image and Video Engineering
Dept. of Electrical & Computer Engr., The University of Texas at Austin,
Austin, Texas 78712-1084, USA.

Abstract

Video transcoding is a key technology to support video communications over heterogeneous networks. Although quite a bit of research effort has been made in video transcoding due to its wide applications, most video transcoding techniques proposed in the literature are optimized based on the simple Mean Squared Error (MSE) metric which does not correlate well with the human visual perception. In this paper, *foveation*, a property of the HVS, is exploited in video transcoding. The proposed foveation embedded DCT domain video transcoding can reduce the bit rate without compromising visual quality or achieve better subjective quality for a given bit rate by shaping the compression distortion according to the foveated contrast sensitivity function of the HVS. In addition, fast algorithms for video foveation filtering and DCT domain inverse motion compensation are developed, which significantly improve the efficiency of video transcoding.

Key words: DCT domain, foveation, MPEG video, video transcoding, video composition.

1 Introduction

With the emergence of video compression standards such as MPEG and H.26x, digital video is becoming widely used in video communications. Meanwhile,

* Corresponding author. Fax: +1-512-471-1225
Email address: shizhongl@yahoo.com; bovik@ece.utexas.edu (Shizhong Liu and Alan C. Bovik).

[1] This work was supported in part by Texas Instruments, Inc. and by the Texas Advanced Technology Program.

the explosive growth of the Internet has created tremendous opportunities for networked multimedia applications such as Video on Demand (VOD), video conferencing and WebTV.

However, video communication over the Internet still faces challenging problems due to the diversity and heterogeneity of the Internet in terms of client device and network connection bandwidth. On the client side, new devices other than traditional desktop computers, such as Personal Digital Assistants (PDA's) and cellular phones, are being used to access the Internet. Different devices usually have different characteristics in terms of display capability, storage capacity, processing power and network access. For instance, handheld computers usually have smaller display screens, memory size and lower processing power, compared to desktop computers. Network connections are also highly diverse, ranging from several kilo-bits per second up to giga-bits per second. With video communication over such a heterogeneous network, adaptation to different client devices and their accessing channel bandwidths is a challenging problem.

Scalable video coding, in which the video source is coded as one base layer and one or more enhancement layers, has been developed in current video coding standards to support heterogeneous video communications. However, the number of enhancement layers supported by the current video compression standards is very limited and no dynamic changes can be done on the compressed video stream during transmission. In addition, the inter-operability between different video coding standards cannot be supported in scalable video coding schemes.

Video transcoding, where video is converted from one compressed format to another compressed format for adaptation of channel bandwidth or receiver or both, is another technique proposed to adaptively deliver video streams across heterogeneous networks. In video transcoding, an incoming video stream is first decoded or partially decoded, then certain operations, such as re-quantization or filtering, are applied to manipulate the decoded video sequence, and finally the manipulated video sequence is re-encoded into a bit stream and sent to the outgoing channel. Converting a video stream to a lower bit-rate version via video transcoding can provide much finer and more dynamic adaptation to various channel situations than using scalable coding schemes. Moreover, with video transcoding, it is also possible to change video format to adapt to different client devices (See Fig. 1), which is impossible in scalable video coding approaches.

Different video transcoding technologies have been proposed in the literature [1–3]. They can be essentially divided into two categories: Pixel domain video transcoding [1] and DCT domain video transcoding [2]. In general, DCT domain video trancoding is more efficient than Pixel domain video transcod-

ing due to the absence of DCT-IDCT operations [2]. One basic problem in video transcoding is how to achieve the optimal visual quality at a given bit rate. In most video applications, human viewers are the final arbiters of video quality. Therefore, Human Vision System (HVS) based video transcoding is desirable to achieve the optimal visual quality at a given bit rate. Nevertheless, most video transcoding techniques proposed in the literature [1, 2] are optimized based on the simple Mean Squared Error (MSE) metric which does not correlate with the HVS very well. In this paper, *foveation*, a property of the HVS, is embedded in video transcoding. Foveation is attributed to the space-variant sampling nature of the HVS, where the resolution is highest within a few degrees of the fixation point, and drops quadratically away from this central region, or fovea, as a function of eccentricity [4]. In the current video compression standards, the foveation feature of the HVS has not been exploited. As a result, most standard video sequences are non-foveated. The proposed foveation embedded video transcoding is to transcode a non-foveated video stream to a foveated one such that the transcoded video stream can be delivered in a lower bandwidth channel with minimum or even no visual degradation under certain viewing configuration. Compared to other video transcoding techniques [1, 2], the foveation embedded video transcoding can reduce the bit rate without compromising visual quality, or can achieve better visual quality at the same bit-rate by shaping the distortion according to the foveation feature of the HVS. Furthermore, fast algorithms for video foveation filtering and DCT domain inverse motion compensation are developed, which improve the efficiency of video transcoding significantly.

The rest of this paper is organized as follows. Section 2 reviews the foveation property of the HVS and its mathematical model. In Section 3, a foveation embedded DCT domain video transcoder is presented. In Section 4, we develop a fast algorithm for DCT domain inverse motion compensation. Section 5 is experimental results and discussions. Finally, we conclude this paper in Section 6.

2 Foveation

2.1 Physiological Aspect of Foveation

It has been found that in the human eye, the photoreceptors (cones and rods) and ganglion cells are not uniformly distributed across the retina, as shown in Fig. 2 [5, 6]. The cones and ganglion cells are very densely packed in the fovea and quickly decrease in density as a function of eccentricity. Due to the non-uniform distribution of photoreceptors and ganglion cells across the retinal, the HVS samples the visual field non-uniformly, where the denser the

photoreceptors, the higher the sampling rate. Thus human visual perception has a space-variant nature where the resolution is highest within a few degrees of the point of fixation, and drops quadratically away from this central region, or fovea, as a function of eccentricity. The highest resolution in the fovea is about 55 cycles/degree. The resolution cutoff is reduced by a factor of two at 2.5 degrees from the point of fixation, and by a factor of ten at 20 degrees [4]. This space-variant characteristic of the HVS is called *foveation*. The location at which the viewer fixates is called the *foveation point*.

Foveation is an effective way for the HVS to compress the information to be transmitted from the retina to the brain by capturing only a small subset of the scene at a high resolution while still maintaining a wide perceptual field. This space-variant structure of the HVS has motivated active research in the design of computer vision and visual communication systems. The foveated image and the uniform full resolution image should be visually indistinguishable under certain viewing configuration, provided that the viewer's fixation point coincides with the foveation point of the foveated image [7]. In principle, this observation can lead to the design of image/video codecs, where large compression gains can be achieved by taking advantage of the foveation characteristic of the human eye without compromising visual quality.

2.2 Foveation Modeling

To take advantage of the foveation characteristic of the HVS in image/video coding systems, it is necessary to know mathematically how the spatial resolution varies as a function of eccentricity in the human eye. Physiological research has provided detailed measurements of the contrast sensitivity of the human eye [8–10]. In [4], Geisler and Perry proposed a contrast threshold formula to fit the human contrast sensitivity data measured as a function of spatial frequency and retinal eccentricity. The formula is

$$CT(f,e) = CT_0 \ \exp(\alpha f \frac{e+e_2}{e_2}) \tag{1}$$

where f is spatial frequency (cycle per degree), e is the retinal eccentricity (degrees), CT_0 is the minimum contrast threshold, α is the spatial frequency decay constant, and e_2 is the half-resolution eccentricity. With $\alpha = 0.106$, $e_2 = 2.3$, and $CT_0 = \frac{1}{76} \sim \frac{1}{64}$, the formula can be fitted to the data measured in [8–10]. From (1), a Foveation Contrast Sensitivity Function (FCSF) can be obtained by defining $FCSF = \frac{1}{CT(f,e)}$ [11]. That is

$$FCSF(f,e) = \frac{1}{CT_0} \exp(-\alpha f \frac{e+e_2}{e_2}). \tag{2}$$

Given an eccentricity e from the foveation point, (1) can be used to find the local maximal perceptual spatial frequency f_c. All spatial frequencies higher than f_c will be invisible in the area beyond the given eccentricity e regardless of their contrast. Specifically, the local cut-off frequency can be found by setting the left side of (1) to 1.0 (the maximum contrast) and solving for f:

$$f_c = \frac{e_2}{\alpha(e+e_2)} ln \frac{1}{CT_0}. \tag{3}$$

Most digital images are obtained by uniform sampling. In order to remove the undetectable high spatial frequencies in an image, it is necessary to map the visual spatial frequency f_c (cycles/degree) to the digital frequency f_d (cycles/pixel) according to the viewing distance. The viewing parameters are shown in Fig. 3, where v is the viewing distance, d is the distance between a pixel and the foveation point, e is the eccentricity of the pixel, and i_p is the image size. Hence

$$e = \frac{180}{\pi} tan^{-1} \frac{d}{v}. \tag{4}$$

Suppose each pixel forms a square with sides of length ϵ. Then

$$\begin{aligned} f_d &= \frac{180}{\pi} [\tan^{-1}(\frac{\epsilon}{2v} + \frac{d}{v}) - \tan^{-1}(\frac{-\epsilon}{2v} + \frac{d}{v})] f_c \\ &\approx \frac{180}{\pi} \frac{1}{1+(\frac{d}{v})^2} \frac{\epsilon}{v} f_c \\ &= \frac{180}{\pi} \frac{\epsilon}{v + \frac{d^2}{v}} f_c. \end{aligned} \tag{5}$$

Since the maximum digital frequency is 0.5, f_d should be bounded by the maximum frequency:

$$f_d = \min[f_d, 0.5]. \tag{6}$$

When $f_d > 0.5$, then the display resolution of the uniform image is already below the highest resolution the human eye can discern, thus no filtering is needed. For images of size 512×512 pixels, the local digital cut-off frequency is plotted against distance from the foveation point, for different viewing distances, in Fig. 4 by assuming the foveation point is at the center of the image. Compression gain can be achieved by filtering out spatial frequencies beyond the cut-off frequency f_d in digital images without causing visual degradation.

In low bit-rate image/video coding systems, it is usually not enough to achieve a given target bit-rate by only removing the spatial frequencies beyond the cut-off frequency f_d in digital images. More distortion may have to be introduced in order to keep the resulting bit-rate from exceeding the target bit-rate. The FCSF in (2) shows that the contrast sensitivity of the human eye declines exponentially as the eccentricity e increases for a fixed spatial frequency, and also declines as the spatial frequency increases at the same eccentricity. Fig. 5 plots the normalized FCSF for several digital frequencies ($f_d = 0.1, 0.2, 0.3, 0.4$), for instance, under the assumption that the viewing distance is three times the image height and the foveation point is at the center of the image. For a given spatial frequency, the same degree of distortion at different locations in the image will yield different visual effects. At the point of fixation, even small errors may cause significant visual distortion since the contrast sensitivity of the eye is highest there. In contrast, large errors in peripheral regions may be invisible due to the reduced contrast sensitivity in those areas. Therefore, it would be advantageous to shape the distortion noise according to the FCSF.

2.3 Foveated Visual Communication Systems

The existence of a space-variant nature in our visual system suggests that a foveated image and the full resolution image is perceptually indistinguishable if the viewer's fixation point coincides with the foveation point of the foveated image. Given a uniform full resolution image, a foveated image can be obtained by matching the spatial resolution of the image to the fall off in spatial resolution of the human eye. In [12], a uniform resolution image was first transformed into log-polar space, then the DCT was applied as the next step in a compression process. Geisler and Perry [4] have proposed the use of an image pyramid representation for foveating images. In a standard image pyramid, as described by Burt and Adelson [13], the input image is low-pass filtered and then down-sampled by a factor of two in both directions to obtain a lower resolution image with one quarter the number of elements. This process of low-pass filtering and down-sampling is repeated to obtain a sequence of successively lower resolution images. To create a foveated version of the original image, they selected regions from each resolution level according to the local maximum spatial resolution which can be detected by the human eye.

For the approaches discussed above, the spatial dimension of the foveated image is usually much smaller than the original (uniform) image. The foveated image is then processed as a uniform image. Due to the reduction of spatial dimension of the foveated image, both computation savings and compression gain can be achieved. However, at the decoding side, corresponding special operations are needed to recover the spatial dimension of the original image

for rendering. For example, in the first approach, the image in the log-polar space has to be transformed back into X-Y space for display. It would be advantageous to allow a standard image/video decoder to decode a foveated image/video without any special operations. In [11, 14], wavelet transforms were employed to foveate images. Using wavelets, an image can be decomposed into four child images. At each level of the wavelet pyramid, four sub-images are created which represent the low- and high- frequency components of the image in each of the two dimensions. Therefore, those portions of the high-frequency sub-images that are far from the foveation point need not be encoded. The resulting foveated image can be decoded by a standard wavelet decoder. In [7], the space-variant resolution of the human eye was mapped to the digital spatial frequency plane. Foveated images were computed by removing undetectable high spatial frequencies via low-pass filtering. The whole image was divided into several regions with different cut-off frequencies which were computed according to (3). Then each region was filtered by a low-pass filter with corresponding cut-off frequency. The foveated image/video can be correctly decoded by a standard image/video decoder such as JPEG, MPEG or H.26x decoder. However, in [7], since the foveation filtering was performed in the spatial domain, the computational complexity corresponding to the convolution of the filter kernel and the image is rather high, which makes this approach not suitable for real-time image/video communications. In this work, we propose a foveation embedded DCT domain video transcoding technique, in which the foveation filtering is performed directly in the DCT domain to reduce the computational complexity of foveation filtering.

3 Foveation Embedded DCT Domain Video Transcoder

The most straightforward way to perform video transcoding is to fully decode the incoming video bitstream and then re-encode the video under new constraints imposed by the outgoing link. However, this approach has high computational complexity and thus low efficiency since it includes both a stand-alone video decoder and a stand-alone video encoder. It is the video transcoder's aim to convey an incoming compressed video bitstream to the outgoing link without the need of fully decoding and re-encoding. This achieves low complexity, low delay and high efficient interconnection of two multimedia networks of similar or diverse types.

In [1], a simple open-loop video transcoder was proposed, in which the incoming bit-rate is down-scaled by truncating the high frequency DCT coefficients or performing a requantization process. Since the transcoding is done in the coded domain, its computational complexity is quite low. However, since the transcoding error associated with the anchor picture is not added to the subsequent inter-coded frames, the transcoding error in the inter-coded

frames will accumulate until the next intra-coded frame is met. This error accumulation is known as *drift*, which results in unacceptable video quality for most applications. Drift-free transcoding is made possible by using a decoder to decode the incoming video and then using an encoder to re-encode the video into another format or at a lower bit rate. However, its high computational complexity makes it difficult to be used in real-time applications. Since a pre-encoded video stream arriving at the transcoder already carries much useful information such as the picture type, motion vectors, quantization step-size, and bit-allocation statistics, it is possible to reduce the complexity of the video transcoder by exploiting some of the available information. By reusing the motion vectors and macroblock coding mode decision information received in the video decoder, fast video transcoders operating in both the pixel domain [3, 15] and the DCT domain [2] can be obtained. Furthermore, it has been shown that DCT domain video transcoding is more efficient than pixel domain video transcoding due to the absence of the DCT-IDCT and the smaller data volume to be processed [2]. However, these fast video transcoders can hardly support either spatial or temporal resolution conversion or video coding format conversions.

While most video transcoding techniques proposed in the literature transcode a uniform resolution video stream to another uniform resolution one at lower bit rate, we propose a fast foveation embedded DCT domain video transcoder illustrated in Fig. 6, in which an incoming uniform resolution video is transcoded into a foveated video to achieve better visual quality [7] by exploiting the foveation property of the HVS. In Fig. 6, The transcoder first decodes the incoming video bit stream to the DCT domain. Then all inter-coded frames are converted to intra-coded frames by the DCT domain Inverse Motion Compensation (IMC). After that, DCT domain foveation filtering is applied to every reconstructed frame. On the encoder side, the motion vectors extracted from the incoming bit stream are used to compute the initial vectors for the outgoing bit stream [16], then a motion vector refinement process is employed to refine the initial motion vectors. Since the motion vector refinement is usually conducted in a small area (*e.g.*, ± 1 pixel [16]) to obtain the optimal vector, a full scale motion estimation, which comprises more than 60 - 70% of the encoding complexity, can be avoided. The proposed video transcoder can support both spatial and temporal as well as video coding format conversions. In the following subsections, DCT domain foveation filtering, DCT domain motion vector refinement, foveated bit rate control and foveation point selection will be discussed. DCT domain inverse motion compensation will be discussed in Section IV.

3.1 DCT Domain Foveation Filtering

In [17], the image is divided into several regions according to eccentricity as shown in Fig. 7. Then each region is low-pass filtered by a 2-D separable FIR filter whose cut-off frequency is derived from (3). Since the filtering is implemented in the pixel domain, the computational complexity is rather high. For example, an N tap spatial filter requires N multiplications and $N-1$ additions for each pixel. Moreover, with the wide acceptance of DCT based image and video compression standards (*e.g.*, JPEG, MPEG), image/video is usually available as a compressed bit stream. To perform foveation over a compressed image, it is usually required to decompress the image to the pixel-domain, then do foveation filtering in the pixel domain, and finally re-compress the foveated image for transmission or storage. Clearly, it would be more efficient to conduct the foveation filtering directly in the DCT domain so that the IDCT-DCT procedure can be avoided.

In this work, we present a DCT domain foveation filtering technique. It has been shown that a simple circular convolution-multiplication relationship for the DCT similar to that for the Discrete Fourier Transform (DFT) exists [18–21]. The multiplication of the DCT of a signal sequence and the DFT of a filter sequence results in circular convolution of the folded signal sequence and the filter sequence, which is called block mirror filtering. Namely, the block mirror filtering in the pixel domain corresponds to the coefficient-by-coefficient multiplication in the DCT domain. Specifically, for 1-D signal, let $X_N(k), k = 0, \ldots, N-1$, be one DCT block data (N is the length of the DCT block and $N = 8$ for image/video coding) and $h(n)$ be an even symmetric FIR filter, *i.e.* $h(-n) = h(n)$. Then

$$Y_N(k) = X_N(k)H_F(k) \quad k = 0, \ldots, N-1 \tag{7}$$

where $Y_N(k)$ is the result of the block mirror filtering in the DCT domain, $H_F(k)$ is the $2N$-point DFT of $h(n)$.

The 1-D block mirror filtering can be easily extended to the 2-D case by using a separable approach [22, 23]. Block mirror filtering implemented in the DCT-domain is simple and easily parallelized since each block is filtered independently. In addition, the coefficient-by-coefficient multiplications in the DCT domain block mirror filtering can be combined with the inverse quantization or quantization process [24, 25] for further reduction of computational complexity. Since neighboring pixels are highly correlated in typical images and no discontinuities are introduced in the block mirroring, an image filtered by the block mirror filtering scheme is close to the result of a true linear convolution [20, 23].

3.2 DCT Domain Motion Vector Refinement

Reusing the motion vectors extracted from the incoming video bit stream usually results in non-optimal video transcoding [26]. In foveation based video transcoding, the incoming video stream is a uniform resolution video sequence while the output is a foveated video sequence. In this case, motion vector refinement is more important than in the uniform resolution case in obtaining the optimal motion vector.

Although the optimal motion vector can be obtained by a new full scale motion estimation, it is not desirable because of its high computational complexity. In the video transcoder, the optimal motion vector can be obtained by refining the incoming motion vector within a relative small range as opposed to applying a full-scale motion estimation [16,26]. While most motion estimation methods proposed in the literature work in the pixel domain, we choose to perform motion vector refinement directly in the DCT domain with consideration of foveation. In pixel domain motion estimation methods, the mean squared error (MSE) and mean absolute difference (MAD) are the most widely used block matching criteria [27]. For DCT domain motion estimation, we can use the following criterion:

$$D(i,j) = \frac{1}{N^2} \sum_{m=1}^{N} \sum_{n=1}^{N} |[u_k(m,n) - u_{k-1}(m+i, n+j)]|. \tag{8}$$

Here, N is the size of each block and $u_k(m,n)$ is the value of the DCT coefficient located at (m,n) in a block of the kth frame. Since the DCT block is usually sparse, the matching computation between two blocks can be reduced, relative to the pixel domain motion estimation. Moreover, if the cut-off frequency of the block $u_k(m,n)$ is f_c, then, we only need to compute the DCT coefficients below f_c. One disadvantage of DCT domain motion estimation is that when the candidate block and target block are not aligned, extraction of the the candidate block from the reference frame in the DCT domain is more complex than in the pixel domain. The fast algorithm for DCT domain inverse motion compensation, discussed in Section 4, can be used to accelerate the extraction process.

3.3 Foveated Bit Rate Control

One major task of video transcoding is to fit the video coding bit rate to the output channel. Usually, the transcoded bit stream has a lower bit rate than the original one does. The objective of optimal video transcoding is to achieve the best visual quality for a given bit rate. Specifically, the optimal video

transcoding is to find a set of quantization parameters for a group of N MB's $\{q_1, q_2, \cdots, q_N\}$ such that the overall distortion D is minimized and the total bit rate R complies with a given target bit rate R_T. This can be fomulated as

$$min \;\; D, \quad subject\ to\ R \leq R_T \tag{9}$$

with D and R given as

$$D = \sum_{k=1}^{N} d_k(q_k) \qquad R = \sum_{k=1}^{N} r_k(q_k)$$

where D is the total distortion, R is the total resulting bit-rate, $d_k(q_k)$ and $r_k(q_k)$ are the distortion and rate of the kth MB corresponding to the quantization parameter q_k.

How to define the distortion measure d_k such that it reflects the actual visual quality of the reconstructed video is a challenging problem. Although the simple Mean Squared Error (MSE) measure is widely used in video encoding or transcoding, it is well known that the MSE measure does not correlate with human visual perception very well. In this work, we propose a foveated distortion measure which is

$$d_k = \sum_{m=1}^{M} \sum_{i=0}^{63} ||FCSF(m,i)(c_o(i) - c_r(i))||_2. \tag{10}$$

Here, M is the number of blocks in one MB; $FCSF(k,i)$ is the foveation contrast sensitivity function of the HVS, determined by the location of the block and the index of the DCT coefficient; c_o and c_r are the original and reconstructed DCT coefficients, respectively. The foveated bit rate control may achieve better visual quality, for a given bit rate, by shaping the distortion according to the FCSF of the HVS.

The constrained problem of (9) can be solved by converting it into the unconstrained problem through a Lagrange multiplier $\lambda \geq 0$. That is

$$J_k(\lambda) = \min\{d_k(q_k) + \lambda\ r_k(q_k)\}. \tag{11}$$

Suppose $(r_k^*(\lambda), d_k^*(\lambda))$ is the solution to the minimum Lagrange cost $J_k(\lambda)$ for MB k, and q_k^* is the corresponding quantizer step size. For any $\lambda \geq 0$, the optimal solution $(R^*(\lambda), D^*(\lambda))$ is the sum of the solutions $(r_k^*(\lambda), d_k^*(\lambda))$ for $k = 1, 2, \cdots, N$. Given $\lambda = \lambda_s$, if the total bit rate happens to be equal to the given bit rate, *i.e.*, $R = R_T$, then the set $\{q_1^*, q_2^*, \cdots, q_N^*\}$ is the optimal quantization parameters. Hence, the optimal value λ_s has to be found for each

group of N MB's. Fast algorithms for searching λ_s have been proposed in the literature [17, 28].

3.4 Foveation Point Selection

In foveated image and video communication systems, finding the location of foveation point(s) is a challenging problem, which explains why the foveation property of the HVS is not exploited in the current image/video coding standards. In [4,7], an interactive method was suggested, where the foveation point is indicated by an eye tracker or other simple pointing device, such as mouse or touch pad, at the receiver side. Then, the location of foveation point is sent back to the sender. Obviously, this method assumes that a reverse channel is available so that the location of the foveation point can be transmitted from the receiver to the sender in a real-time fashion. However, in some applications such as TV broadcasting, the reverse channel is not available or has a long delay. The other way is to automatically find likely foveation point(s) at the sender side by analyzing the video data. The foveation point(s) may be determined by using algorithms that identify regions of interest in the video sequence based on object segmentation, edge information and contrast or texture information [29–34]. It is difficult to develop a generic algorithm to automatically locate the foveation point(s) in various video sequences. However, in some specific applications, it may become feasible to locate the foveation point(s). For example, in video conferencing or news broadcasting applications, human faces are usually the primary objects in the video sequences and the face regions are very likely the locations that observers are to fixate at. Therefore, in this case, the problem of finding the foveation point(s) is equivalent to that of human face detection. In this work, we assume that the location of foveation point(s) is known.

4 DCT Domain Inverse Motion Compensation

The problem of *DCT-domain inverse motion compensation* was studied by Chang et al. [23]. The general setup is shown in Fig. 8, where $\hat{x}$ is the current block of interest, x_1, x_2 x_3 and x_4 are the reference blocks from which $\hat{x}$ is derived. According to [23], $\hat{x}$ can be expressed as a superposition of the appropriate windowed and shifted versions of x_1, x_2 x_3 and x_4, *i.e.*,

$$\hat{x} = \sum_{i=1}^{4} q_{i1} x_i q_{i2} \tag{12}$$

where $q_{ij}, i = 1, \ldots, 4, j = 1, 2$ are sparse 8×8 matrices of zeros and ones that perform windowing and shifting operations. For example, for $i = 1$,

$$q_{11} = \begin{pmatrix} O & I_h \\ O & O \end{pmatrix}, \; q_{12} = \begin{pmatrix} O & O \\ I_w & O \end{pmatrix}, \tag{13}$$

where I_h and I_w are identity matrices of dimension $h \times h$ and $w \times w$, respectively. The values h and w are determined by the motion vector corresponding to $\hat{x}$. By applying the distributive property of matrix multiplication with respect to DCT [23], one can obtain its DCT domain counterpart as

$$\hat{X} = \sum_{i=1}^{4} Q_{i1} X_i Q_{i2} \tag{14}$$

where $\hat{X}$, X_i, Q_{i1} and Q_{i2} are the DCT's of $\hat{x}$, x_i, q_{i1} and q_{i2}, respectively. Be noted that the matrices Q_{i1} and Q_{i2} are constant hence can be pre-computed and stored in memory [23].

Brute-force computation of (14) in the case where the reference block $\hat{x}$ is not aligned in any direction with the block structure requires eight floating-point matrix multiplications and three matrix additions. Several algorithms have been proposed to reduce the computational complexity of the DCT-domain inverse motion compensation [2, 35–37]. For example, in [35], Merhav *et al.* proposed to factorize the constant matrices Q_{ij} into a series of relatively sparse matrices instead of fully pre-computing them. As a result, some of the matrix multiplications in (14) can be replaced by simple addition and permutation operations such that computational complexity can be reduced. Assunção *et al.* [2] approximated the elements of Q_{ij} by binary numbers with a maximum distortion of $\frac{1}{32}$ so that all multiplications can be implemented by basic integer operations such as *shift* and *add*. They showed that in terms of operations (*shift*, *add*) required, their algorithm has only 28% of the computational complexity of the method proposed by Merhav *et al.* [35] while the distortion introduced by the approximation is negligible (about 0.2 dB as reported in [2]).

In this work, we explain a novel technique to speed-up the DCT domain inverse motion compensation. While most algorithms proposed in the literature focus on how to reduce the computational complexity of (14) via matrix factorization or approximation, we approach the problem from a different angle by analyzing the statistical properties of natural image/video data. By modeling a natural image as a 2-D separable Markov Random Field [38], we estimate the local bandwidth of the target block to be reconstructed from the reference blocks. The algorithm can reduce the processing time by avoiding the computations of

those DCT coefficients outside the estimated local bandwidth. To compute the DCT coefficients inside the estimated local bandwidth, other fast algorithms proposed in the literature such as [2,23,35,36] can be employed. Experimental results show that the proposed algorithm achieves computational improvement of 25% to 55% without visual degradation, compared to Chang's algorithm in [23].

4.1 The Basic Idea

As discussed, inverse motion compensation consists of two basic operations, *i.e., windowing and shifting*. The *windowing* operation keeps the data inside the window unchanged but zeros all data outside the window. As a result, it usually introduces a steep change at the edge of the window, which means that many artificial high frequency components are possibly introduced by the algorithm. To clarify, let us study the 1-D case. Fig. 9 shows a narrow-band signal $y(n)$ obtained by summing two functions, *i.e.*, $y(n) = y_l(n) + y_r(n)$, $0 \leq n < N$. Let $w_l(n), w_r(n)$ be two window functions, *i.e*,

$$w_l(n) = \begin{cases} 1,\ 0 \leq n \leq M \\ 0,\ otherwise \end{cases} \text{ and } w_r(n) = \begin{cases} 1,\ M < n < N \\ 0,\ otherwise. \end{cases}$$

We can write $y_l(n) = y(n)w_l(n)$ and $y_r(n) = y(n)w_r(n)$. Let $Y(e^{j\omega}), Y_l(e^{j\omega}), Y_r(e^{j\omega}), W_l(e^{j\omega})$ and $W_r(e^{j\omega})$ be the Discrete Time Fourier Transforms of $y(n), y_l(n), y_r(n), w_l(n)$ and $w_r(n)$, respectively. Then the following equations can be obtained:

$$Y_l(e^{j\omega}) = Y(e^{j\omega}) \otimes W_l(e^{j\omega}) \tag{15}$$

$$Y_r(e^{j\omega}) = Y(e^{j\omega}) \otimes W_r(e^{j\omega}) \tag{16}$$

where $\otimes$ denotes convolution of two periodic functions with the limits of integration extending over only one period. We also have

$$|W_l(e^{j\omega})| = \frac{\sin[\omega(M+1)/2]}{\sin(\omega/2)}. \tag{17}$$

Let B, B_l, B_r, B_w^l and B_w^r be the bandwidths of $y(n), y_l(n), y_r(n), w_l(n)$ and $w_r(n)$, respectively. From (17), we can roughly estimate $B_w^l \approx \frac{2\pi}{M+1}$, M being the length of the window. B_w^r has similar format as B_w^l. From (15) and (16), we can obtain the following inequalities:

$$B_l > \max(B, B_w^l) \tag{18}$$

$$B_r > \max(B, B_w^r). \tag{19}$$

Let E_l be the frequency components beyond B in B_l, and E_r be the frequency components beyond B in B_r. Since $y(n) = y_l(n)+y_r(n)$, the following equation must be satisfied

$$E_l + E_r = 0. \tag{20}$$

This means that all frequency components beyond B will disappear after summation, implying that there is no need to compute them. Therefore, if we can estimate the frequency bandwidth B of $y(n)$ before constructing $Y(e^{j\omega})$, we need only compute those frequency components inside B when computing $Y_l(e^{j\omega})$ and $Y_r(e^{j\omega})$. This is the basic idea of the proposed algorithm described in the next subsection.

4.2 Local Bandwidth Constrained Inverse Motion Compensation

Generally, neighboring pixels are highly correlated in images. This inter-pixel correlation is often modeled by using Markov Random Field (MRF) models [39]. In [38], Sikora *et al.* also assumed that the 2-D image random field is separable with identical and stationary correlation along each image dimension and that the simple first order AR(1) Markov model was adopted to model the pixel-to-pixel correlation along image rows and columns. For each image row, the variance-normalized AR(1) 1-D auto-correlation function can expressed as

$$R_x = \alpha^{|n|}, \tag{21}$$

where n describes the distance between two images pixels and α denotes the pixel-to-pixel correlation in the row. α typically takes values ranging from 0.9 to 0.98 [38,40]. Fig. 10 shows two 1-D eight point adjacent blocks L_1 and L_2 in an image row. According to the above model, L_1 and L_2 should have the same power spectral density function, hence the same bandwidth because they have the same correlation function [41]. Similarly, if we want to extract L_3 (shown in Fig. 10) from L_1 and L_2, we can predict that L_3 also has the same bandwidth as L_1 and L_2 based on the model. However, images are usually non-stationary, so the bandwidth of L_1 is often different from that of L_2. To account for this, we take the maximum bandwidth as the estimate for L_3, *i.e.*,

$$B_3 \approx \max(B_1, B_2), \tag{22}$$

where B_1, B_2 and B_3 are the bandwidth of L_1, L_2 and L_3, respectively. For example, if the maximum index of the non-zero DCT coefficients (here we use DCT coefficients as the representations of frequency components) is 2 in L_1 and 4 in L_2, we estimate that the maximum index of the non-zero DCT coefficients in L_3 is 4. To extract the DCT coefficients directly from the DCT's of L_1 and L_2, we only need to compute those DCT coefficients with index no greater than 4 in L_3. 2-D problem can be easily converted into two 1-D problems by using separable implementations.

4.3 Accuracy of Local Bandwidth Estimation

As discussed, image/video data is usually a non-stationary random signal. To account for this, the maximum bandwidth of two adjacent blocks is taken as the estimate of the bandwidth of the block to be extracted. However, estimation error still exists under certain circumstances. For example, in Fig. 10, assume L_1 and L_2 are both constant blocks but there is a discontinuity at the boundary between the two blocks. According to the proposed algorithm, the block L_3 should also be a constant block since both L_1 and L_2 only have DC component in the frequency domain. However, the block L_3 actually contains a step discontinuity. The probability of such kind of estimation error will be higher in the images containing lots of edge information than in the relatively smooth images. In addition, since each block in the frame is independently quantized by certain quantization factor, the correlation between adjacent blocks is reduced, which may also make the local bandwidth estimation inaccurate. Several monochrome images with the dimension of 512×512 have been selected to examine the accuracy of the proposed method for local bandwidth estimation in real images. In the experiment, if the estimated bandwidth is smaller than the actual bandwidth of the target block, the estimation is considered incorrect. Otherwise, the estimation is correct. The results are shown in Fig. 11. It can be seen that more than 97% of the estimations are correct for all quantization parameters. The correctness of estimation declines as the quantization increases, implying that more distortion would be introduced in the image/video with large quantization parameter.

5 Experimental Results

A foveation embedded DCT domain video transcoder was implemented. The proposed video transcoder can also be changed to a non-foveated video transcoder by turning off the foveation module. In the experiments, we first evaluate the performance of the fast algorithm for DCT domain inverse motion compensation proposed in this paper; then we estimate the bit rate reduction due to

Table 1
Average time to convert one P or B frame to one I frame at the bit rate of 1 Mb/s (Unit: Seconds)

Video Sequence	P frame		B frame	
	Original method	Proposed method	Original method	Proposed method
Foreman	0.2512	0.1324	0.3987	0.2152
Coastguard	0.1912	0.0937	0.3099	0.1490
Mobile	0.2983	0.1550	0.3686	0.2061
Stefan	0.1636	0.0743	0.2941	0.1408

foveation; and finally we compare the visual quality of foveated video with that of non-foveated video encoded at the same bit rate.

5.1 Performance of Fast DCT Domain Inverse Motion Compensation

The method proposed by Chang and Messerschmitt [23] was implemented as the original algorithm to evaluate the performance of the proposed method. For comparison, both methods were integrated into the non-foveated DCT domain video transcoder as the inverse motion compensation module, respectively. The input of the transcoder was a MPEG-coded video bit-stream with the frame rate of 30 frames per second. The GOP structure of the encoded video is $M = 3, N = 12$, *i.e.*, IBBPBBPBBPBB. We transcoded all P and B frames in the incoming bit-stream back to I frames by the DCT domain inverse motion compensation. Since the proposed algorithm only computes those DCT coefficients inside the estimated bandwidth, we first investigate the distortion caused by the algorithm by comparing the PSNR values of those I frames recovered from P or B frames using both methods, respectively. Then we measure the computing time of both methods to show the computational improvement of the proposed algorithms. Four video sequences *Foreman*, *Coastguard*, *Mobile* and *Stefan* were selected and encoded at the bit rate of 1 Mb/s in the experiments. The PSNR results for each frame after inverse motion compensation are shown in Fig. 12. The average PSNR degradation is 0.29 dB in *Foreman*, 0.35 dB in *Coastguard*, 0.51 dB in *Mobile* and 0.36 dB in *Stefan*. The PSNR degradation depends on the images being tested. For example, in the sequence *mobile*, the pictures have a lot of strong edges and are very dynamic; hence the AR model of our algorithm is inaccurate, which increases the error probability of local bandwidth estimation as discussed in Section 4. As a result, the PSNR of *mobile* degrades more than that of other sequences. The average computing time for reconstructing a P or B frame to one I frame is listed in Table 1. The computing time is measured on a Windows NT workstation with 512MB memory and 300MHz

Table 2
Bit rate reduction due to foveation.

Foreman	*Coastguard*	*Mobile*	*Stefan*	*Akiyo*
35.4%	37.7%	39.8%	32.6%	30.4%

Pentium II Processor (32K non-blocking, level-one cache, and 512K unified, non-blocking, level-two cache.). Relative to the original method, the proposed algorithm achieves 45 - 55% computing time savings.

5.2 Bit Rate Reduction Due to Foveation

The video sequence *news* with CIF resolution (352×288) is used in the experiments. The viewing distance is assumed two times the image height and the foveation point is on the man's face. The original video sequence is encoded at 1 Mb/s in MPEG2 format format, which is a high quality video. Then, the encoded video bit stream is transcoded to a H.263 video bit stream using constant quantization parameter $Q = 10$. Fig. 13 shows the resulting bits for each coded frame with and without foveation, respectively. As can be seen, the average bit rate reduction due to foveation is more than 35%. However, the visual quality of the foveated video is almost the same as that with the uniform resolution under the assumed viewing condition, as shown in Fig. 14. The bit rate reductions for other video sequences have similar results which are listed in Table 2.

5.3 Comparison of Visual Quality

In foveated video transcoding, foveation can also be used to shape the encoding distortion according to the foveated contrast sensitivity function (FCSF) of the HVS for better visual quality. To show that, we transcode the video bit stream from 1 Mb/s to 80 Kb/s using the foveated video transcoding and non-foveated video transcoding methods, respectively. Fig. 15 shows the 20th and 40th frames of the resulting video sequences. Note that the man's face area in the foveated video has higher visual quality than that in the uniform resolution one, whereas other areas in the foveated video are worse than the corresponding areas in the non-foveated video. However, when the viewer perceives the video with the viewing distance two times the image height and fixates at the man's face area, the severe distortion in the peripheral area is much less perceptually visible than that at the fove. Hence, the foveated video exhibits better perceptual quality under the assumed viewing configuration. Fig. 16 shows the 60th and 80th frames of the resulting video sequences, in which the foveation point is set at the woman's face in the foveated video. Sim-

ilarly, the woman's face area in the foveated video has higher visual quality than that in the uniform resolution video. Therefore, when the viewer fixates at the woman's face area, the foveated video has better perceptual quality.

6 Conclusion

In this paper, we have demonstrated a foveated video trancoding technique by exploiting the foveation property of the HVS. The proposed foveated video transcoder encodes the foveal area with higher quality than the peripheral area to match the fall-off of the foveated contrast sensitivity function (FCSF) of the HVS. Experimental results have shown that the proposed foveated video transcoding technique can reduce bit rate without compromising visual quality or achieve better visual quality at a given bit rate than other video transcoding techniques, provided that the observer fixates at the foveation point. The proposed foveated video transcoding results in fully standard compatible bit streams, so no modification is required at the receiver side. Moreover, we developed fast DCT domain image foveation technique and fast DCT domain inverse motion compensation algorithm to significantly improve the efficiency of the video transcoding. The proposed techniques are especially useful in very low bit rate video communications.

References

[1] H. Sun, W. Kwok, and J. W. Zdepski, "Architectures for MPEG compressed bitstream scaling," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 6, pp. 191–199, Apr. 1996.

[2] P. A. A. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 8, pp. 953–967, Dec. 1998.

[3] K.-S. Kan and K.-C. Fan, "Video transcoding architecture with minimum buffer requirement for compressed MPEG-2 bitstream," *Signal Processing*, vol. 67, pp. 223–235, 1998.

[4] W. S. Geisler and J. S. Perry, "A real-time foveated multiresolution system for low-bandwidth video communication," in *SPIE Proceedings*, vol. 3299, pp. 294–305, July 1998.

[5] B. A. Wandell, *Foundations of Vision.* Sunderland, MA: Sinauer Associates, Inc., 1994.

[6] T. Caelli, *Visual Perception Theory and Practice.* New York, NY: Robert Maxwell, M.C., 1981.

[7] S. Lee, *Foveated Video Compression and Visual Communications over Wireless and Wireline Networks*. PhD thesis, Dept. of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX, May 2000.

[8] J. G. Robson and N. Graham, "Probability summation and regional variation in contrast sensitivity across the visual field," *Vision Research*, vol. 21, pp. 409–418, 1981.

[9] M. S. Banks, A. B. Sekuler, and S. J. Anderson, "Peripheral spatial vision: limits imposed by optics, photoreceptors, and receptor pooling," *Journal of the Optical Socity of America*, vol. 8, pp. 1775–1787, 1991.

[10] T. L. Arnow and W. S. Geisler, "Visual detection following retinal damage: Prediction of an inhomogeneous retino-cortical model," in *SPIE Proceedings: Human Vision and Electronic Imaging*, vol. 2674, pp. 119–130, 1996.

[11] Z. Wang and A. C. Bovik, "Embedded foveation image coding," *IEEE Trans. on Image Processing*, vol. 10, pp. 1397–1410, Oct. 2001.

[12] C. Weiman, "Video compression via a log polar mapping," in *SPIE Proceedings: Real time image processing II*, vol. 1295, pp. 266–277, 1990.

[13] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. on Comm.*, vol. 31, pp. 532–540, 1983.

[14] E.-C. Chang, *Foveation Techniques and Scheduling Issues in Thinwire Visualization*. PhD thesis, Dept. of Computer Science, New York University, New York, NY, May 1998.

[15] G. Keesman, R. Hellinghuizen, F. Hoeksema, and G. Heideman, "Transcoding of MPEG bitstreams," *Signal Processing: Image Communication*, vol. 8, pp. 481–500, 1996.

[16] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," *IEEE Trans. on Multimedia*, vol. 2, pp. 101–110, June 2000.

[17] S. Lee, S. Pattichis, and A. C. Bovik, "Foveated video compression with optimal rate control," *IEEE Trans. on Image Processing*, vol. 10, pp. 977–992, July 2001.

[18] H.-Y. Kim and R. Meyer, "DCT domain filter for ATV down conversion," *IEEE Tran. on Consumer Electronics*, vol. 43, pp. 1074–1078, Nov. 1997.

[19] K. N. Ngan and R. J. Clarke, "Lowpass filtering in the cosine transform domain," in *Proc. IEEE Int. Conf. on Comm.*, (Seattle, WA), pp. 31.7.1–31.7.5, June 1980.

[20] B. Chitprasert and K. R. Rao, "Discrete cosine transform filtering," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 3, pp. 1281–1284, Apr. 1990.

[21] W. H. Chen and S. C. Fralick, "Image enhancement using cosine transform filtering," in *Image Sci. Math. Symp.*, (Montrey, CA), Nov. 1976.

[22] J. B. Lee and B. G. Lee, "Transform domain filtering based on pipelining structure," *IEEE Trans. on Signal Processing*, vol. 40, pp. 2061–2064, Aug. 1992.

[23] S.-F. Chang and D. G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video," *IEEE J. on Selected Areas in Comm.*, vol. 13, pp. 1–11, Jan. 1995.

[24] H. R. Sheikh, "Real-time foveation techniques for low bit rate video coding," Master's thesis, Dept. of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78731, May 2001.

[25] H. R. Sheikh, S. Liu, B. L. Evans, and A. C. Bovik, "Real-time foveation techniques for H.263 video enoding in software," in *Proc. IEEE Int. Conf. on Accoustics, Speech and Signal Proc.*, May 2001.

[26] J. Youn, M.-T. Sun, and C.-W. Lin, "Motion vector refinement for high-performance transcoding," *IEEE Trans. on Multimedia*, vol. 1, pp. 30–40, Mar. 1999.

[27] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard*. New York, NY: Chapman & Hall, 1997.

[28] Y. Shohman and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 1445–1453, Sept. 1988.

[29] C. M. Privitera and L. W. Stark, "Algorithms for defining visual regions-of-interest: Comparison with eye fixations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 970–982, Sept. 2000.

[30] H.-L. Eng and K.-K. Ma, "Segmentation and tracking of faces in color images," in *Proc. IEEE Int. Conf. Image Proc.*, vol. 3, pp. 758–761, 2000.

[31] A. Bors and I. Pitas, "Prediction and tracking of moving objects in image sequences," *IEEE Trans. on Image Processing*, vol. 8, pp. 1441–1445, Aug. 2000.

[32] H. Zen, T. Hasegawa, and S. Ozawa, "Moving object detection from MPEG coded picture," in *Proc. IEEE Int. Conf. Image Proc.*, vol. 4, pp. 25–29, 1999.

[33] H.-L. Eng and K.-K. Ma, "Motion trajectory extraction based on macroblock motion vectors for video indexing," in *Proc. IEEE Int. Conf. Image Proc.*, vol. 3, pp. 284–288, 1999.

[34] L. Favalli, A. Mecocci, and F. Moschetti, "Object tracking for retrieval applications in MPEG-2," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 10, pp. 427–432, Apr. 2000.

[35] N. Merhav and V. Bhaskaran, "Fast algorithm for DCT-domain image down-sampling and for inverse motion compensation," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 7, pp. 468–476, June 1997.

[36] J. Song and B.-L. Yeo, "A fast algorithm for DCT-domain inverse motion compensation based on shared information in a macroblock," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 10, pp. 767–775, Aug. 2000.

[37] S. Acharya and B. Smith, "Compressed domain transcoding of MPEG," in *Proc. of the International Conference on Multimedia Computing and Systems*, vol. 4, (Austin, TX), pp. 25–28, June 1998.

[38] T. Sikora and H. Li, "Optimal block-overlapping synthesis transforms for coding images and video at very low bitrates," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 6, pp. 157–167, Apr. 1996.

[39] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 5, pp. 25–39, Jan. 1983.

[40] I. M. Pao and M. T. Sun, "Modeling DCT coefficients for fast video encoding," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 9, pp. 608–616, June 1999.

[41] M. H. Hayes, *Statistical Digital Signal Processing and Modeling.* New York, NY: John Wiley & Sons, Inc., 1996.

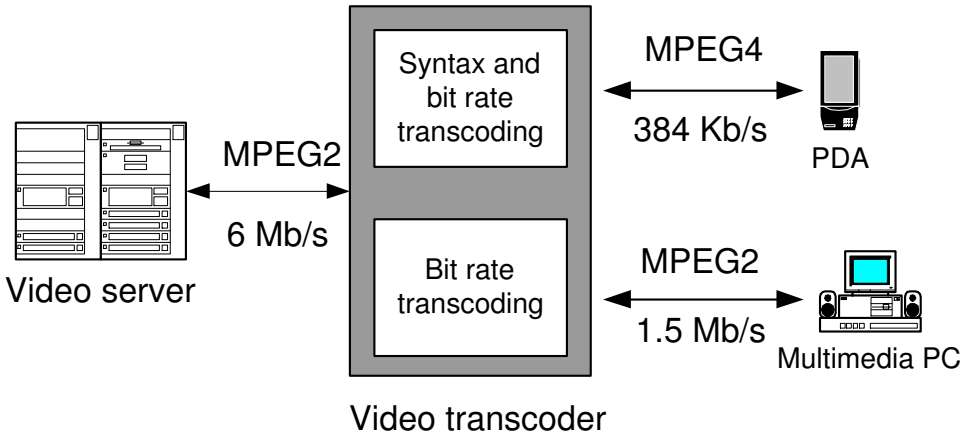
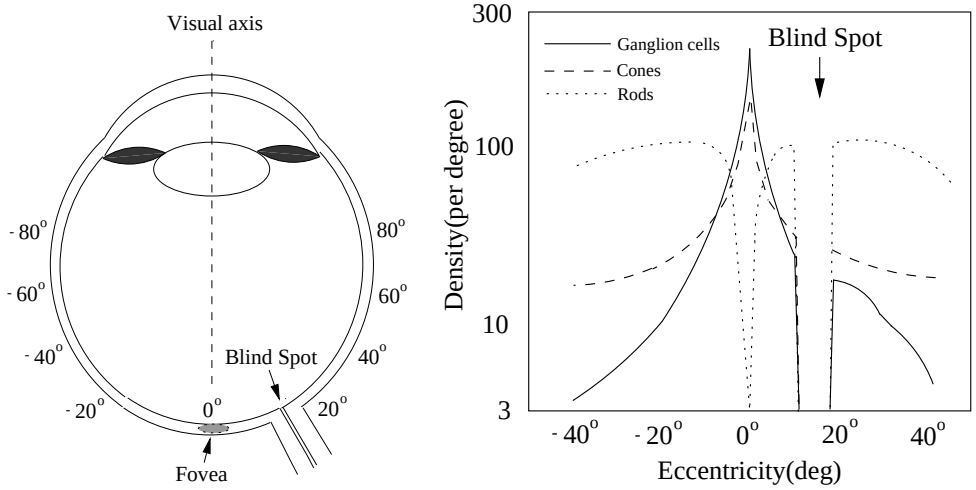


Fig. 1. Illustration of video transcoding.



(a) Eccentricity.

(b) Density of cones, rods and ganglion cells

Fig. 2. The human eye and associated photoreceptors (rods and cones), and ganglion cells distributions across the retina. (a) Eccentricity — Degree of visual angle relative to the position of fovea for the left eye. (b) The distribution of photoreceptors (cones and rods) and ganglion cells across the retina as a function of eccentricity.

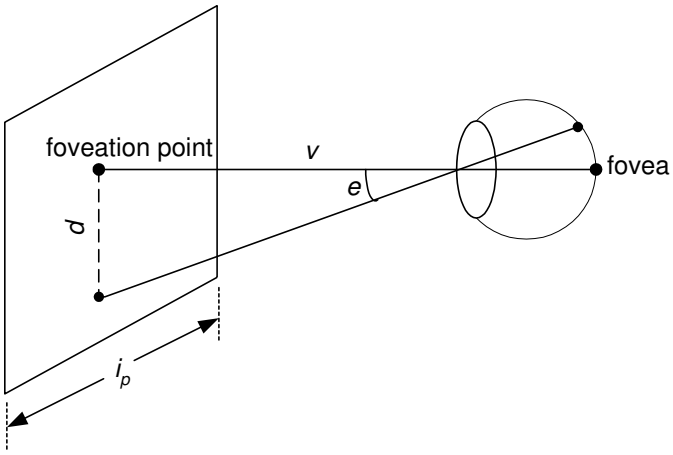


Fig. 3. Viewing parameters

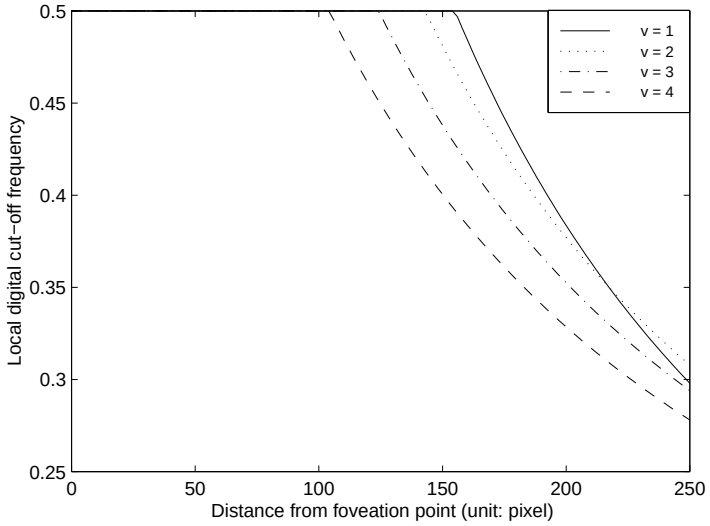


Fig. 4. Local digital cut-off frequency of foveation filter with $CT_0 = \frac{1}{74}$. The unit of viewing distance v is the image height measured in pixel (in this case, the image height is 512 pixels).

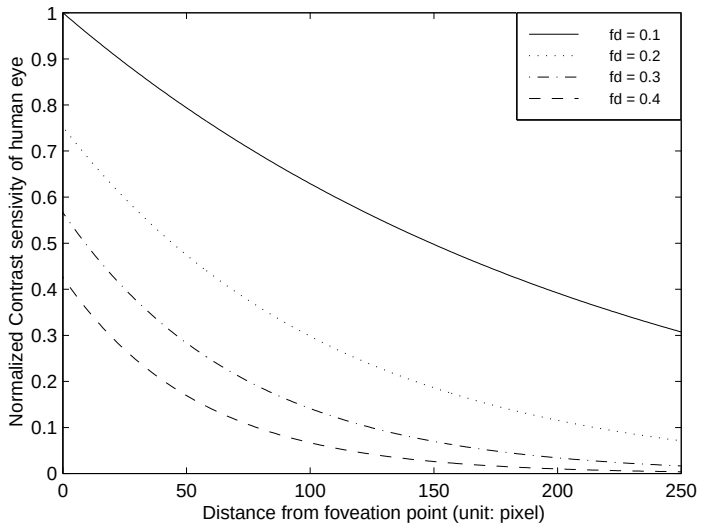


Fig. 5. Normalized foveation contrast sensitivity at different digital frequencies f_d.

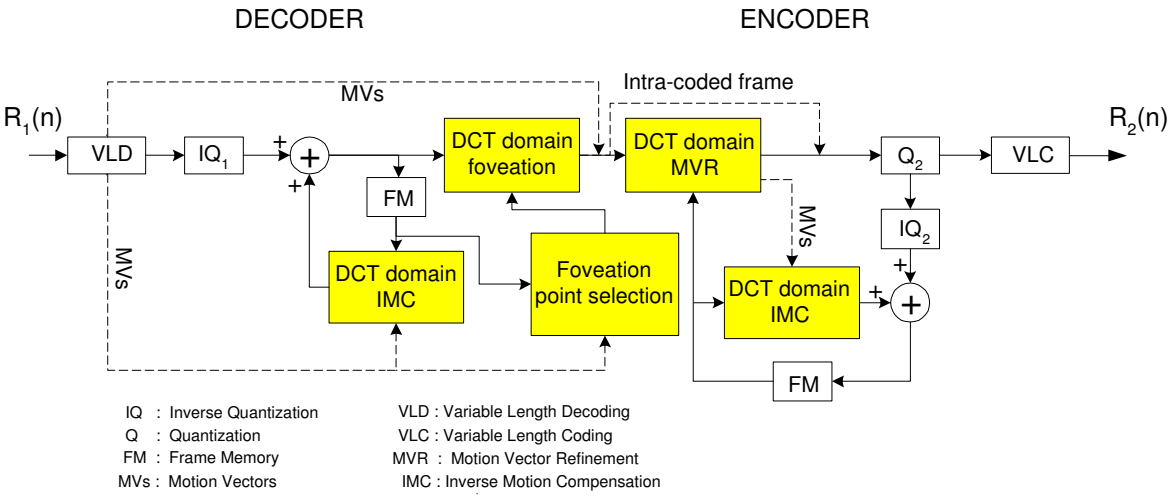


Fig. 6. Foveation embedded DCT domain video transcoder.

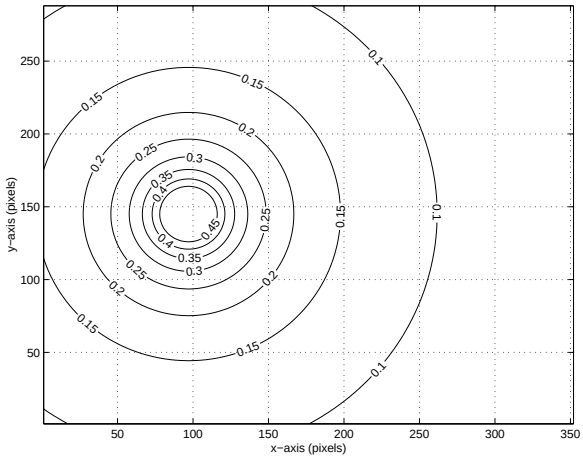


Fig. 7. Illustration of dividing a image into several regions with different cut-off frequencies.

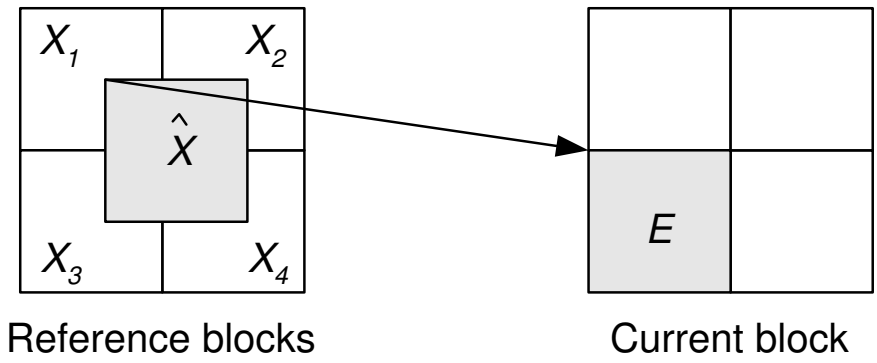


Fig. 8. DCT domain inverse motion compensation.

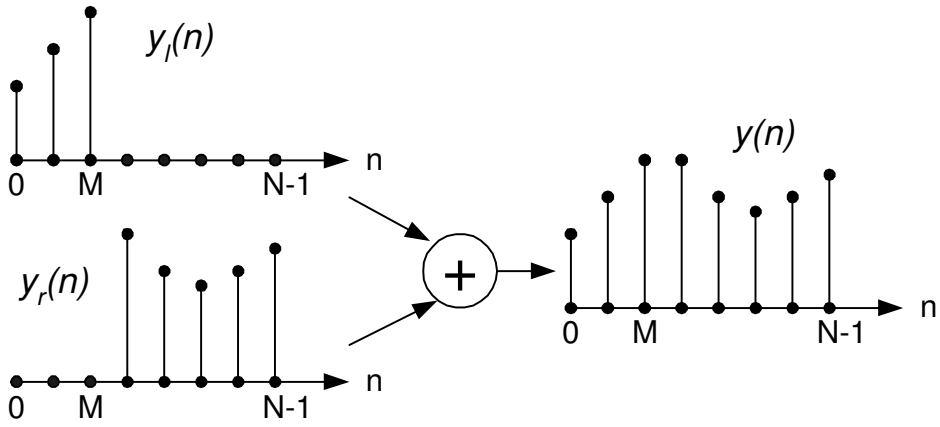


Fig. 9. 1-D windowing operation.

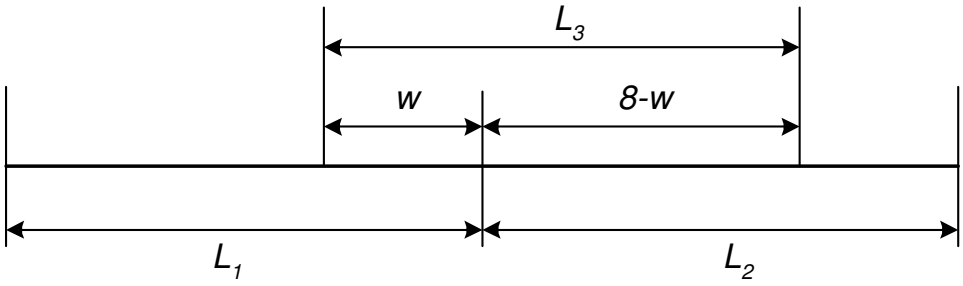


Fig. 10. 1-D block extraction.

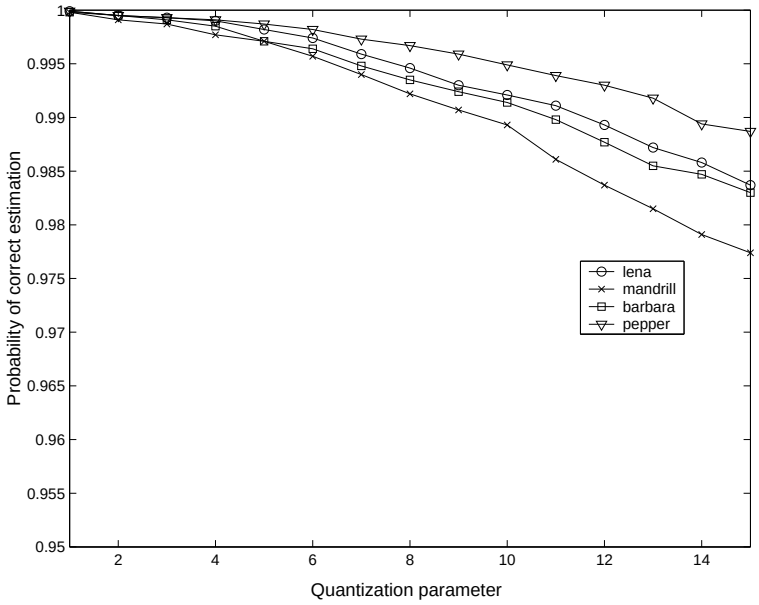
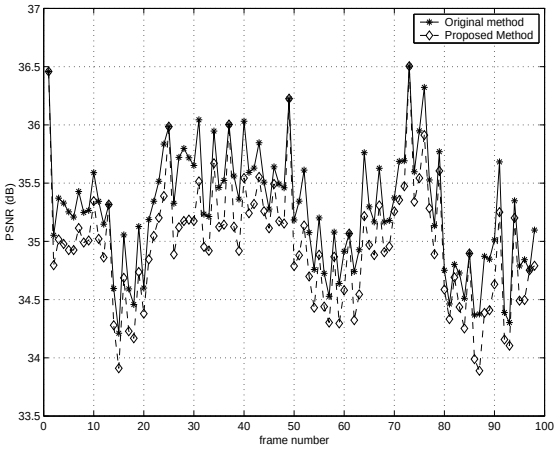
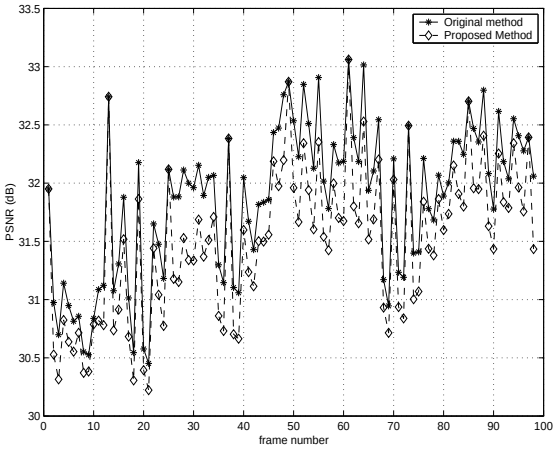


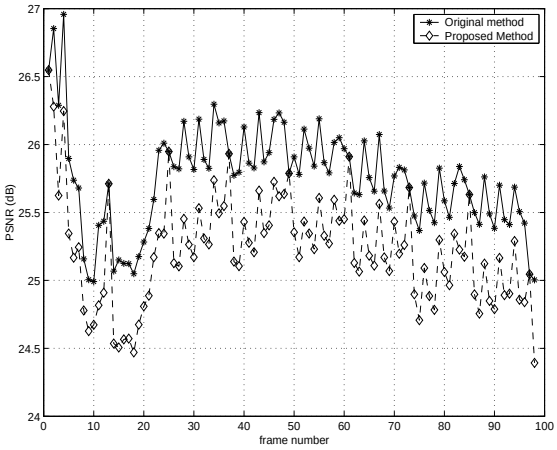
Fig. 11. Accuracy of local bandwidth estimation.



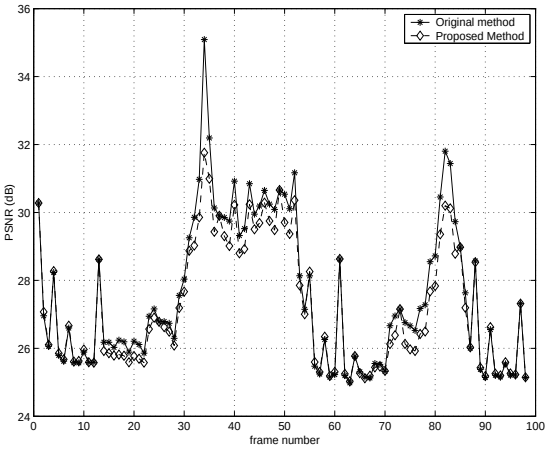
(a) PSNR of *Foreman* at 1 Mb/s.



(b) PSNR of *Coastguard* at 1 Mb/s.



(c) PSNR of *Mobile* at 1 Mb/s.



(d) PSNR of *Stefan* at 1 Mb/s.

Fig. 12. PSNR value of each frame reconstructed by different inverse motion compensation algorithms.

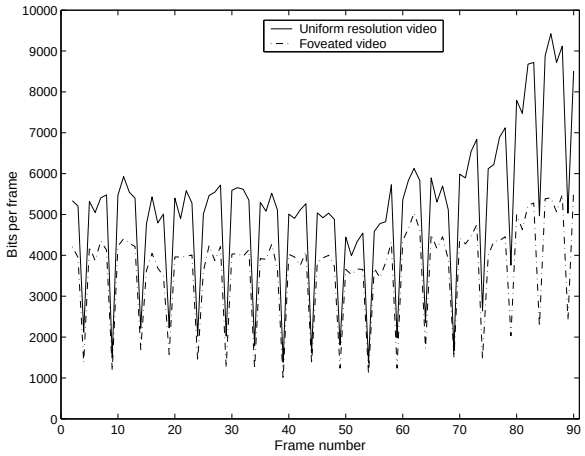


Fig. 13. The number of bits for each frame.

(a) Uniform resolution video. (b) Foveated video.

Fig. 14. Visual quality of transcoded video with $Q = 10$.

(a) The 20th frame of the uniform resolution video.

(b) The 20th frame of the foveated video (Foveation point is at the Man's face area).

(c) The 40th frame of the uniform resolution video.

(d) The 40th frame of the foveated video (Foveation point is at the Man's face area).

Fig. 15. Visual quality of transcoded video at the target bit-rate of 80Kb/s.

(a) The 60th frame of the uniform resolution video.

(b) The 60th frame of the foveated video (Foveation point is at the Woman's face area).

(c) The 80th frame of the uniform resolution video.

(d) The 80th frame of the foveated video (Foveation point is at the Woman's face area).

Fig. 16. Visual quality of transcoded video at the target bit-rate of 80Kb/s.