Transactions Letters

A Scheme for Spatial Scalability Using Nonscalable Encoders

Rakesh Dugad, *Member, IEEE*, and Narendra Ahuja, *Fellow, IEEE*

***Abstract*—We describe a scheme that achieves spatially scalable coding of video by employing *nonscalable* video encoders (e.g., MPEG-2 main profile), along with a downsampler and an upsampler. The scheme is illustrated for the case of coding video at two resolutions. The enhancement layer is coded in two steps by first exploiting the spatial redundancy and then exploiting the temporal redundancy. Hence, the scheme has a separable implementation. Results are presented for five different sequences, coded for three different combinations of base and enhancement layer bit rates. When MPEG-2 main profile is used for the nonscalable encoders, the results obtained are comparable to the performance of MPEG-2 spatial scalability profile.**

***Index Terms*—Discrete cosine transform (DCT), HDTV, MPEG-2, multiresolution coding, scalable video compression, spatial scalability, subband decomposition.**

I. Introduction

To ACCOMMODATE the varied requirements on computational speed, bandwidth, and compatibility with existing equipment, many applications require that a compressed video stream be decodable at various resolutions and signal qualities. Of the various ways of achieving such scalable compression, we shall focus on *spatial scalability*.

In spatial scalability, the video is coded at a hierarchy of spatial resolutions with each higher layer using the (decoded) lower layers for spatial prediction [1]. In case of two resolutions, the lower layer is called the *base layer* and the higher layer is called the *enhancement layer*. Hence, to obtain the video at lower resolution, only the base layer need be decoded, but to get the higher resolution video, both the base and enhancement layers need to be decoded and combined. A special case is *simulcast* in which the video at each of the various resolutions is coded independently of the video at every other resolution. This is wasteful of bandwidth because the bitrate can be reduced by exploiting the redundancy across various resolutions as in spatial scalability.

Spatial scalability has many applications. It is used in HDTV to maintain compatibility with standard definition TV. For transmitting video over dual-priority networks, we can transmit a low-resolution version of the video over high-priority channel and an enhancement layer over the low-priority channel. Also, one solution to transmitting video over bandwidth-constrained channels is to transmit a low-resolution version of the video. For browsing a remote video database, it would be more economical to send low-resolution versions of the video clips to the user and then, depending on his or her interest, progressively enhance the resolution.

In this paper, we propose a scheme that achieves spatial scalability by using two *nonscalable* encoders (e.g., MPEG-2 main profile), along with a downsampler and an upsampler. Achieving the functionality of spatial scalability with standard equipment that already contains a number of nonscalable encoders is economically and practically very attractive.

A. Overview of Previous Work

A scheme for scalable compression of images using Laplacian pyramid was first proposed by Burt and Adelson [2]. Later, the subband decomposition of images [3], [4], along with the theory of wavelets [5], removed the redundancy present in the pyramid representation. Very efficient schemes, such as the embedded zero-tree wavelet (EZW) algorithm of Shapiro [6] and the procedure of set partitioning in hierarchical trees (SPIHT) introduced by Said and Pearlman [7] for such scalable compression based on subband decomposition, have been devised for still image compression.

However, the extension of such schemes to video is not straightforward because exploiting temporal redundancy usually involves recursive prediction (in the temporal direction). This implies that the encoder and the decoder have to maintain the *same* state (prediction value) to avoid error propagation. Hence, if the decoder is able to only partially decode the stream, its state will not match with that of the encoder. This leads to error propagation, also called *drift*.

Various schemes based on two-dimensional (2-D) [8]–[14] and three-dimensional (3-D) [15]–[18] subband decompositions, with and without motion compensation, have been proposed. These schemes extend the ideas from scalable image compression (e.g., subband decomposition) and nonscalable video compression (e.g., motion compensation) to achieve scalable video compression and avoid the problem of drift.

Drift can also be eliminated by coding the video explicitly at various resolutions, while exploiting the redundancies across the resolutions to reduce the bit rate. For example, in the case of two resolutions, a base layer is created (with any nonscalable scheme) containing the video at lower resolution. To create the enhancement layer, the high-resolution frame is predicted

Manuscript received October 4, 2001; revised October 25, 2002. This work was supported by the Office of Naval Research under Grant N00014-96-1-0502. This paper was recommended by Associate Editor J.-R. Ohm.

R. Dugad was with the Department of Electrical and Computer Engineering and Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA. He is now with Flarion Technologies, Bedminster, NJ 07921 USA (e-mail: dugad@vision.ai.uiuc.edu).

N. Ahuja is with the Department of Electrical and Computer Engineering and Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: ahuja@vision.ai.uiuc.edu).

Digital Object Identifier 10.1109/TCSVT.2003.816519

1051-8215/03$17.00 © 2003 IEEE

by predicting either its pixel values or its transform coefficients. The spatial scalability scheme adopted in MPEG-2 predicts the pixel values using a weighed combination (on a macroblock by macroblock basis) of an upsampled version of the low-resolution frame, and a motion-compensated version of the previous reference frame. This allows for coding the video at two different bit rates. The fine granularity scalability (FGS) adopted in MPEG-4 allows for coding the video at a variety of bit rates [19]. The enhancement layer codes the difference between the original and the picture reconstructed from the base layer using bitplane coding of discrete cosine transform (DCT) coefficients. Unlike MPEG-2, recursive temporal prediction is not used in the enhancement layer which can be truncated into any number of bits per picture after encoding is complete. The enhancement layer quality is proportional to the number of bits decoded.

The transform coefficients of the enhancement layer can be predicted in the following manner while exploiting temporal redundancy [10], [11], [13]. A motion-compensated prediction of the current high-resolution frame is formed by replacing each block by its prediction (closest match in MSE sense) in the previous decoded high-resolution reference frame. This motion-compensated prediction is decomposed with discrete wavelet transform (DWT), and the high-frequency coefficients are used as prediction for the corresponding high-frequency coefficients of the subband decomposition of current high-resolution frame. The residual (prediction error) thus obtained is quantized and coded directly [11], or first DCT transformed and then quantized and coded [10], [12]. This approach has the following two problems.

- The low-frequency components of a block play a crucial role in deciding its motion-compensated match in the previous reference frame. However, only the high-frequency components are predicted after motion compensation. Since most blocks have significant energy in the low-frequency components, motion compensation will be not very effective at minimizing the energy in the residual of the high-frequency components.
- If the predicted block contains parts which are shifted versions of the corresponding parts of original block, then prediction of the high-frequency components will suffer (because the DWT is shift-variant).

B. Motivation for the Proposed Scheme

In the above section, we saw the problems associated with predicting the high-frequency coefficients of the 2-D subband decomposition of a frame by motion compensation on the spatial-domain frames (which have predominantly low-frequency content). In our scheme, we use motion compensation to directly match the high-frequency contents which are to be coded in the enhancement layer. This is achieved by performing motion compensation on a spatial representation of the high-frequency contents (e.g., edges in the high-resolution frame).

Another feature of our scheme is that it employs two nonscalable encoders, along with a downsampler and an upsampler. Achieving the functionality of spatial scalability with equipment containing a number of nonscalable encoders is economically and practically very attractive. Such functionality provides an affordable path to high-definition broadcasting while maintaining compatibility with existing standards and equipment. For example, various HDTV encoders in the market employ six SDTV encoders (which already exist on their standard equipment) to get the effect of an HDTV encoder [20], [21]. This makes the equipment for SDTV encoding more attractive to the broadcasters, who can use the same equipment for HDTV broadcasting when they are ready for it. Such equipment also allows for switching between SDTV and HDTV transmissions.

Our scheme works with any nonscalable encoders. However, we will illustrate our scheme for the case of MPEG-2 encoder, which is widely used for encoding high-definition video. The scheme does not require any special hardware apart from a downsampler, an upsampler, and two nonscalable encoders. The scheme works in a sequential fashion by first exploiting the spatial redundancy and then exploiting the temporal redundancy on a frame-by-frame basis. Also, there are no weights to be chosen for combining the spatial and temporal predictions. The scheme is described in Section III.

C. Organization of the Paper

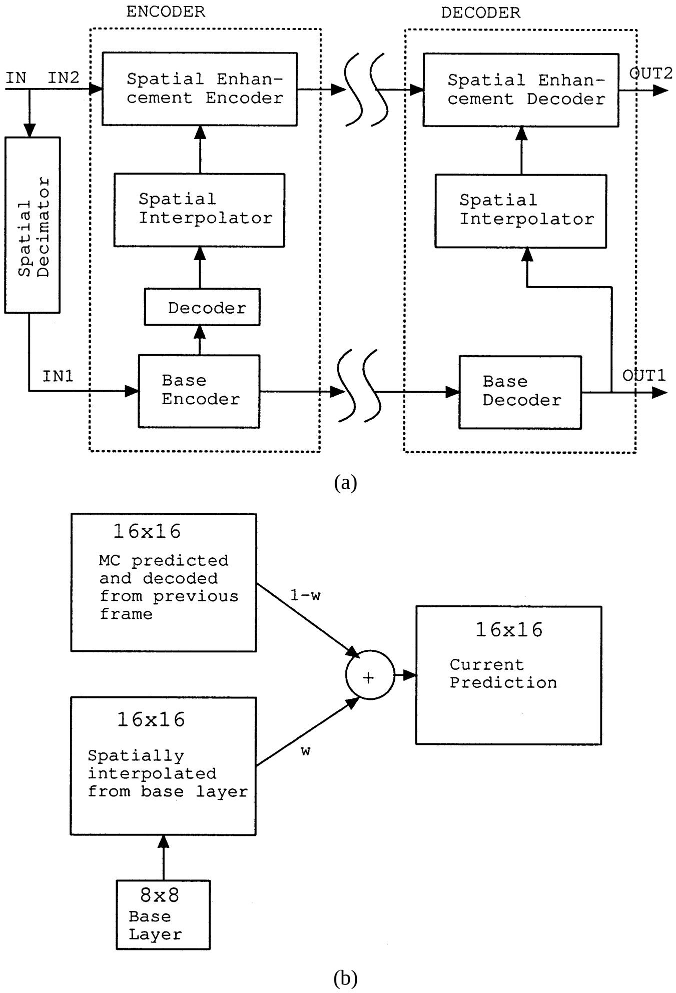
Section II describes the spatial scalability scheme used in MPEG-2. Section III describes our scheme for spatial scalability. Section IV describes the downsizing and upsizing schemes used in MPEG-2 and our DCT-based schemes. Section V presents our results for several sequences. Conclusions are presented in Section VI.

II. Spatial Scalability in MPEG-2

A block diagram of the MPEG-2 spatial scalability scheme [22] is shown in Fig. 1(a). We only consider the case in which the video is coded at two different spatial resolutions, the higher resolution being double the size of the lower resolution in each direction. Each frame is downsampled (the downsampling scheme need not be standardized) to produce the lower resolution frames. These frames are coded using a nonscalable scheme,[1] e.g., the main profile of MPEG-2. The compressed stream containing the video at the lower resolution is called the base layer.

Now, consider how the enhancement layer is created. As shown in Fig. 1(b), the macroblock in the current frame is predicted using a convex linear combination of two macroblocks. The first macroblock is the motion-compensated macroblock of the current macroblock [the MC macroblock can be obtained from the most recently decoded full-resolution frame (P frames), or from a combination of past and future reference macroblocks (B frames), or it can be simply a uniform block of grayscale 128 (I frames)]. This macroblock serves to exploit temporal redundancy. The second macroblock is obtained by upsampling the corresponding 8×8 block in the current decoded low-resolution frame. This macroblock serves to exploit spatial redundancy.

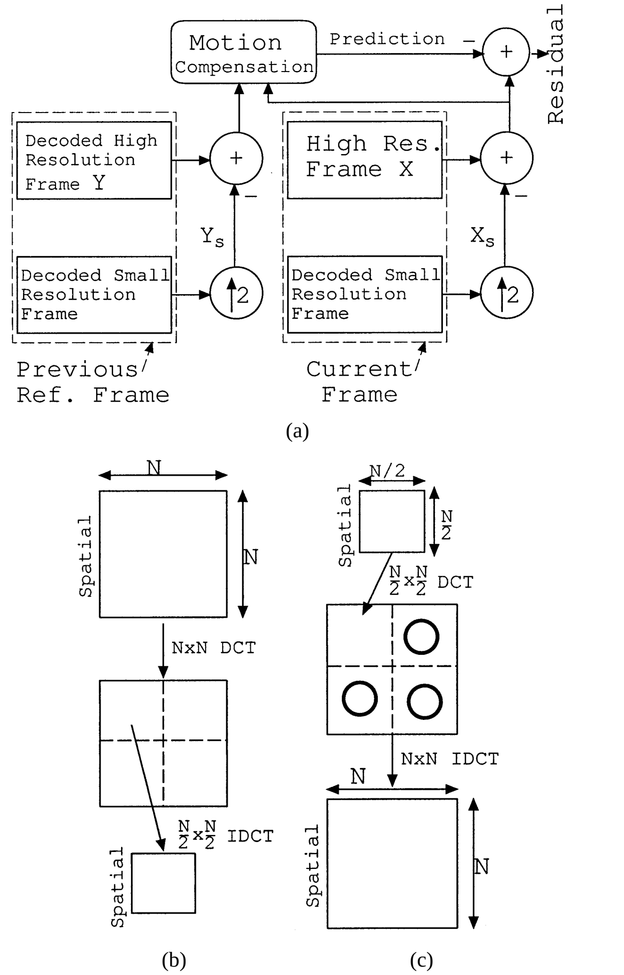
[1] The lower resolution frames could be coded with any standard, not necessarily MPEG-2, because to code and decode the higher resolution frames, we only need to be able to code and decode the lower resolution frames; the details of this coding/decoding need not be known to decode the higher resolution frames. In fact, this is how one can maintain compatibility with other standards using spatial scalability.



(a)

(b)

Fig. 1. (a) Spatial scalability scheme in MPEG-2 [1]. (b) Forming the spatiotemporal prediction for current macroblock in MPEG-2.



(a)

(b)

(c)

Fig. 2. (a) Our spatial scalability scheme. (b) DCT downsampling scheme. (c) DCT upsampling scheme.

In the spatial scalability profile of MPEG-2, the linear combining weight w is allowed to have only three values, viz., $w \in \{0, 0.5, 1\}$. Further, these values cannot be used arbitrarily for all macroblocks. In the case of interlaced video, two weights (w_1, w_2) are allowed per macroblock, one per field. Hence, for each macroblock one has to choose from nine pairs. This would require four bits per macroblock to represent the choice of weights. However, in MPEG-2, various combinations of these pairs are put in four tables. Each table contains a maximum of four pairs and one table is allowed per picture (frame). For example, table 01 contains the pairs (0, 1), (0, 0.5), (0.5, 1), and (0.5, 0.5). Hence, if we decide to use table 01 to code the current frame, then we have to choose one pair from these four pairs per macroblock of the current frame (i.e., all nine pairs are not possible).

Making an optimal choice of table per picture and then choosing an optimal weight pair (from the table) per macroblock of the picture can be computationally very intensive.

III. Proposed Scheme

The MPEG-2 spatial scalability scheme attempts to exploit both the spatial and temporal redundancy at the same time, i.e., once the weight for the spatial prediction is decided, the weight for the temporal prediction is fixed, and *vice versa*. Further, as noted before, finding a globally optimum weight is computationally very expensive. We propose a scheme that first exploits the spatial redundancy and then exploits the temporal redundancy without any weights to be chosen.

The basic idea of our scheme for the case of two resolutions (generalization to include more resolutions is straightforward) is illustrated in Fig. 3 [see also Fig. 2(a) for notation and details]. There are four steps.

1) Decide on a downsizing and an upsizing scheme.
2) Create a low-resolution version of the video using the downsizing scheme on each frame. Code the low-resolution video with a nonscalable encoder, like MPEG-2, to get the base layer. The enhancement layer is created in the next two steps.
3) A spatial prediction X_s of the current frame is formed by upsampling the low-resolution (decoded) frame, and the residual (error) frame $(X - X_s)$ is calculated. Hence, this spatial residual represents the new information in the current frame with respect to its lower resolution version. It mainly consists of edges in the high-resolution frame. The same procedure is repeated at the previous reference frame (using decoded versions of high and low-resolution frames) to get its spatial residual $(Y - Y_s)$.

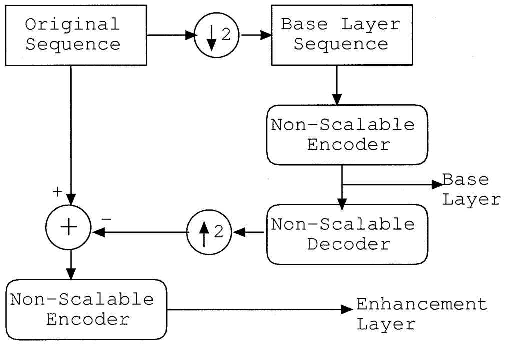


Fig. 3. Alternate representation of our scheme shown in Fig. 2(a).

4) The temporal correlation between these two residuals is exploited by predicting the current spatial residual ($X - X_s$) from the previous spatial residual ($Y - Y_s$) using block motion compensation.

In this manner, we have also removed the temporal redundancy. The resulting residual (which is actually a residual of the spatial residual) is DCT transformed, quantized, and coded as in MPEG-2. We see that one nonscalable MPEG-2 encoder for the base layer and one for the enhancement layer, along with a downsampler and an upsampler, suffice to produce the spatially scalable stream. It can also be expected that a block of spatial residual of current frame will match a block of spatial residual of previous frame at approximately the same shift as the corresponding spatial high-resolution blocks would match. Hence, the motion vectors used in the base layer can be upscaled to predict the motion vectors for the enhancement layer.

Motion compensation is known to be very effective at removing temporal redundancy in a video sequence. However, motion compensation is not very effective on the frequency domain coefficients (which are obtained by using a transform that is typically shift variant). In our scheme, motion compensation for generating the enhancement layer is carried out on the spatial-domain residual images. The spatial-domain residual image ($X - X_s$ or $Y - Y_s$) corresponds to a spatial representation of the high-frequency components of the corresponding high-resolution frame. In other words, we can view the residual image as the inverse transform of the high-frequency components of the high-resolution frame, while the low-frequency components are zeroed out. The residual images are temporally predicted using motion compensation to get the final residual images which are (DCT) transformed, quantized, and coded. Being able to employ motion compensation in this manner yields very effective compression.

If we use the downsampling and upsampling schemes described in [23], we can preserve all the low-frequency components of current high-resolution frame, i.e., the spatial residual consists merely of the high-frequency coefficients of each block (ignoring the effect of quantization).

Our scheme, as described in this section, is called the *decoupled* scheme, because it decouples the exploitation of spatial and temporal redundancies for spatially scalable compression.

IV. Downsizing and Upsizing Schemes

The decoupled scheme described above could be used with any downsizing and upsizing schemes. In Section IV-A, we describe the DCT-based scheme that was proposed in [23]. Section IV-B describes the resizing schemes used in MPEG-2. Section V reports results for all four possible cases (using our decoupled or MPEG-2 spatial scalability scheme, with MPEG-2 or DCT-based resizing scheme).

A. DCT-Based Downsizing and Upsizing Schemes

We shall only consider the case in which the high-resolution frames are twice the size of the small-resolution frames in each direction. Let the high-resolution frames be divided into $N \times N$ blocks (we use $N = 8$ in our results). Each of this $N \times N$ block is independently downsized to $N/2 \times N/2$ block, as shown in Fig. 2(b). Basically, the $N \times N$ block is transformed using DCT, and the $N/2 \times N/2$ low-frequency coefficients are inverse transformed using $N/2 \times N/2$ IDCT to get a spatial block, which is a downsized version of the original $N \times N$ block. The upsampling scheme is exactly the reverse of the downsampling scheme. A given small-size image is divided into $N/2 \times N/2$ blocks, and each block is transformed using $N/2 \times N/2$ DCT. The transformed block constitutes low-frequency coefficients of an $N \times N$ DCT block whose high-frequency coefficients are made equal to zero, as shown in Fig. 2(c). Hence, the spatial prediction $\mathbf{X}_s$ of $\mathbf{X}$ contains all the low-frequency coefficients of $\mathbf{X}$: in this sense, the prediction $\mathbf{X}_s$ captures *all* the spatial redundancy between $\mathbf{X}$ and its downsized version available in the base layer. Fast and compressed domain implementation of these schemes is provided in [23].

When the enhancement and base layers are each in interlaced format (as in interlace–interlace spatial scalability), the procedure described above is applied to each field individually. When they are in a progressive format (as in progressive–progressive spatial scalability), we apply it to each frame.

B. MPEG-2 Downsizing and Upsizing Schemes

First, consider the case of interlace–interlace scalability where both the enhancement layer and the base layer are to be coded in an interlaced format [1]. Also, for now consider only the luminance component.

The downsampling scheme, which is not part of the MPEG-2 standard, is implemented in the following three steps as outlined in [1]: 1) deinterlacing of each field; 2) horizontal downsampling by two; and 3) vertical downsampling by four for each field. Deinterlacing refers to interpolating the samples corresponding to the other field. Thus, when deinterlacing the top field, we interpolate samples corresponding to the bottom field (so that the top field now becomes the size of a frame). The filter used is $(-1/16, 1/2, 1/8, 1/2, -1/16)$, which is also used during upsizing in MPEG-2. Here, the tap 1/8 multiplies the row in the bottom field that is being interpolated in the top field. Horizontal downsampling (i.e., filtering followed by dropping every other column) of each row is then carried out using the odd-length filter $(-29, 0, 88, 138, 88, 0, -29)/256$. Vertical

TABLE I
PSNR of Luminance Component With Base and Enhancement Layers Coded at Various Bit Rates. Each Sequence has Frame Size of 720 × 480 and Frame Rate of 30 Frames/s. the PSNRs Reported are Average Values Over 150 Frames. the First Four Columns are for the Full-Resolution Frames and the Last Two Columns for the Base-Layer Frames. the Numbers in Brackets Denote the Improvements Over the PSNR Values for mpeg_wt (The MPEG-2 Spatial Scalability Scheme) or Over the Base MPEG PSNR Values. Detailed Explanation of Notation is Given in Section V.
(a) Base Layer at 2.0 Mbits/s and Enhancement Layer at 3.0 Mbits/s.
(b) Base Layer at 2.5 Mbits/s and Enhancement Layer at 3.5 Mbits/s.
(c) Base Layer at 4.0 Mbits/s and Enhancement Layer at 6.0 Mbits/s.

Seq	mpeg_wt	dct_wt	dct_dc	mpeg_dc	Base MPEG	Base DCT
Football	33.4	33.9 (+0.5)	33.1 (-0.3)	32.6 (-0.8)	34.6	35.1 (+0.5)
Bike	26.8	27.3 (+0.6)	26.6 (-0.2)	26.1 (-0.7)	28.5	29.2 (+0.7)
Cheers	28.0	28.5 (+0.5)	27.5 (-0.5)	27.1 (-0.9)	27.6	28.5 (+0.9)
Tt	29.8	30.2 (+0.4)	29.6 (-0.2)	29.3 (-0.5)	36.9	36.5 (-0.3)
Susie	40.5	40.9 (+0.4)	40.1 (-0.4)	39.4 (-1.1)	42.4	43.2 (+0.8)

(a)

Seq	mpeg_wt	dct_wt	dct_dc	mpeg_dc	Base MPEG	Base DCT
Football	34.3	34.9 (+0.6)	34.2 (-0.1)	33.6 (-0.7)	36.1	36.5 (+0.5)
Bike	27.6	28.3 (+0.6)	27.6 (-0.0)	27.0 (-0.6)	29.8	30.4 (+0.6)
Cheers	28.9	29.5 (+0.6)	28.6 (-0.3)	28.1 (-0.9)	28.9	29.8 (+0.9)
Tt	30.5	30.9 (+0.4)	30.4 (-0.1)	30.0 (-0.5)	38.2	37.8 (-0.4)
Susie	40.9	41.3 (+0.4)	40.5 (-0.3)	39.8 (-1.1)	43.3	44.1 (+0.9)

(b)

Seq	mpeg_wt	dct_wt	dct_dc	mpeg_dc	Base MPEG	Base DCT
Football	37.1	37.6 (+0.6)	37.0 (-0.1)	36.4 (-0.7)	39.2	39.6 (+0.4)
Bike	30.5	31.3 (+0.7)	30.7 (+0.1)	30.0 (-0.6)	33.0	33.6 (+0.6)
Cheers	31.9	32.7 (+0.8)	31.9 (+0.0)	31.0 (-0.9)	32.0	32.9 (+0.9)
Tt	32.6	33.1 (+0.4)	32.7 (+0.0)	32.2 (-0.4)	41.1	40.8 (-0.4)
Susie	42.1	42.6 (+0.5)	41.8 (-0.3)	41.1 (-1.0)	45.2	45.9 (+0.7)

(c)

downsampling is implemented in two steps: first, downsample by the odd-length filter $(-29, 0, 88, 138, 88, 0, -29)/256$, and then downsample again by the same filter. The center of the odd-length filter is made to coincide with the position of the rows corresponding to the top field. Since the vertical filter lengths used are odd, this will give us a row situated at alternate locations, corresponding to the rows of the top field. A similar procedure is adopted for the bottom field except that in the last step of vertical downsampling, we first downsample by the odd-length filter given before, but then downsample by the *even*-length filter given by $(-4, 23, 109, 109, 23, -4)/256$. This way, each row corresponding to the downsized bottom field will lie at the center of the corresponding two rows of the top field. Hence, the downsized frame is also interlaced.

Upsampling is standardized and is carried out as described next. First, consider upsampling the top field. This is first deinterlaced as described above in downsampling to double its vertical size. Horizontal size is doubled by simple averaging in horizontal direction [using the filter $(0.5, 0.5)$]. This gives the top field of the upsized frame. For the bottom field, the procedure is similar except that before horizontal interpolation, the deinterlaced field is resampled at the midpoint of its rows (by averaging adjacent rows). This positions the deinterlaced field rows at the center of the corresponding rows of the top field.

The same procedure is followed for the chrominance samples except that the deinterlacing filter is simple averaging $(0.5, 0.5)$, i.e., deinterlacing of the top field is done by averaging its adjacent rows to double the number of rows. Bottom-field values are not used when interpolating the top field, and similarly for the bottom field.

Now, consider progressive–progressive scalability where both the base and enhancement layers are coded in progressive format (as frames rather than fields). The downsizing (for both luminance and chrominance components) is accomplished by downsampling in horizontal and vertical directions, each using the odd-length filter $(-29, 0, 88, 138, 88, 0, -29)/256$. The upsizing, which is part of the standard, is accomplished by bilinear interpolation with filter $(0.5, 0.5)$ in both the horizontal and vertical directions.

V. Results

We have used the MPEG-2 encoder provided by the MPEG Software Simulation Group [24]. This implementation provides the MPEG-2 nonscalable codec, but implementation for spatial scalability is not provided. We modified their code to implement the coding of the enhancement layer. For this implementation, the motion vectors for the enhancement layer were obtained independently of the motion vectors for the base layer. Also, we allow all of the nine possible pairs (one per field) of weights (see Section II) for each macroblock. MPEG-2 allows four out of nine possible pairs (one per field) of weights (see Section II) for predicting each macroblock of a given frame. Our implementation of MPEG-2 spatial scalability allows all the nine pairs. This can only improve the performance of MPEG-2 scalability, and hence, we would be comparing our results with the best performance possible with MPEG-2.

We have provided results on five sequences: Football, Cheers (Cheerleaders), Tt, Bike, and Susie. Each sequence has a frame size of $720\,(W) \times 480\,(H)$ and frame rate of 30 frames/s. The motion-vector search ranges for Football, Cheers, Tt, and Susie sequences were chosen as the 95% probability search range (containing 95% of the motion vectors) reported in [25]. The group of pictures (GOP) consists of 15 frames, and the distance between reference frames is three. Hence, the GOP structure is IBBPBBPBBP…

The sequences Football, Cheers, Tt, and Bike are coded in interlace–interlace spatial scalability mode and Susie is coded with progressive–progressive mode [1] (see Section IV for details).

Results are shown in Table I for the luminance component at various bit rates for the base and enhancement layers. The first four columns show results for the full resolution frames and the last two columns for the base layer frames. In the expressions mpeg_wt, dct_wt, dct_dc, and mpeg_dc, the first word denotes the downszing/upsizing scheme used: either the MPEG-2 scheme as described in Section IV-B, or the DCT-based scheme as described in Section IV-A. The second word denotes the spatial scalability scheme being used: either the weighted (wt) scheme used in MPEG-2 (as described in Section II), or our decoupled (dc) scheme described in Section III. Hence, mpeg_wt refers to the spatial scalability scheme in MPEG-2, whereas dct_wt refers to the MPEG-2 scheme but with the downsizing and upsizing operations replaced with our DCT-based downsizing and upsizing schemes.

We see that dct_wt performs best and the performance of dct_dc is close to that of mpeg_wt. The perceptual quality of the frames is about the same. Note that the improvement of dct_wt over mpeg_wt increases with the bit rates of the base and enhancement layers. Note that dct_wt is simply the MPEG-2 spatial scalability scheme with bilinear downsizing and upsizing schemes replaced by the DCT-based downsizing and upsizing schemes described in Section IV-A and also in [23]. The performance of dct_dc becomes comparable to that of mpeg_wt as the bit rates are increased. The scheme mpeg_dc performs worse than mpeg_wt. This highlights the importance of using the DCT-based downsizing and upsizing schemes, which preserve the low-frequency DCT coefficients in the spatial prediction.

The Base MPEG and Base DCT columns show the peak signal-to-noise ratio (PSNR) for the decoded base-layer frames when using the MPEG-2 nonscalable scheme to code the original base-layer frames. For the Base MPEG column, the original base-layer frames are created by downsampling the full-resolution frames using a bilinear scheme as in MPEG-2 (described in Section IV-B). For Base DCT, the downsampling scheme is DCT based, as described in Section IV-A. Thus, Base MPEG shows the base-layer PSNR for the mpeg_wt and mpeg_dc schemes, and Base DCT that for dct_wt and dct_dc. With the DCT-based scheme, the PSNR is usually (except for the Tt sequence) better by over 0.5 dB compared with the MPEG-based downsampling scheme. The perceptual quality of the frames is about the same.

VI. Conclusion

In this paper, we introduced a scheme for spatially scalable coding of video by employing two nonscalable video encoders along with a downsampler and an upsampler. PSNR results were presented for five sequences for three different combinations of base-layer and enhancement-layer bit rates. The scheme works in a sequential manner. It first codes the small-resolution frames (downsampled versions of large-resolution frames), and then codes the difference between the original high-resolution frames and their spatial predictions derived from the decoded small-resolution frames. Both the small-resolution frames and the difference frames can be coded using a nonscalable encoder such as the main profile of MPEG-2. Hence, there are no weight tables or weights to be chosen to exploit the spatial and temporal redundancies. Further, we use motion compensation on the difference frames since these frames are represented in spatial domain. Compatibility with an existing standard like MPEG-2 can be maintained by coding the base layer with that standard. When the MPEG-2 main profile is used for the nonscalable encoders and the DCT-based (described in Section IV-A) scheme is used for downsizing and upsizing, the PSNR performance of our scheme is typically within 0.3 dB of the MPEG-2 scalability scheme, and the perceptual quality is the same as for the MPEG-2 spatial scalability scheme.

We also presented results for a scheme (dct_wt) that simply replaces the bilinear downsampling and upsampling schemes in the MPEG-2 spatial scalability scheme with the DCT-based schemes described in Section IV-A. Typically, this yields 0.5-dB improvement in PSNR, though the perceptual quality of decoded frames is about the same.

References

[1] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An Introduction to MPEG-2*. New York: Chapman Hall, 1997, Digital Multimedia Standards Ser..

[2] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, pp. 532–540, Apr. 1983.

[3] J. Woods and S. O'Neil, "Subband coding of images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1278–1288, Oct. 1986.

[4] A. Tran and K.-M. Liu, "An efficient pyramid image coding system," in *Int. Conf. Acoustics, Speech and Signal Processing*, 1987, pp. 744–747.

[5] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 674–693, July 1989.

[6] J. Shapiro, "Embedded image coding using zero-trees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.

[7] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, June 1996.

[8] S. A. Martucci, I. Sodagar, T. Chiang, and Y.-Q. Zhang, "A zero-tree wavelet video coder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 109–118, Feb. 1997.

[9] K. Shen and E. J. Delp, "Wavelet based rate scalable video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 109–122, Feb. 1999.

[10] U. Benzler, "Scalable multiresolution video coding using subband decomposition," in *Proc. 1st Int. Workshop Wireless Image/Video Communications*, 1996, pp. 109–114.

[11] T. Yoshida and K. Sawada, "Spatio-temporal scalable video coding using subband and adaptive field/frame interpolation," in *Proc. IEEE Asia Pacific Conf. Circuits and Systems*, 1996, pp. 145–148.

[12] M. Domanski, A. Luczak, and S. Mackowiak, "Spatio-temporal scalability for MPEG video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 1088–1093, Oct. 2000.

[13] T. Naveen and J. W. Woods, "Motion compensated multiresolution transmission of high definition video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 29–41, Feb. 1994.

[14] J. W. Woods and G. Lilienfield, "A resolution and frame-rate scalable subband/wavelet video coder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 1035–1044, Sept. 2001.

[15] J.-R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Trans. Image Processing*, vol. 3, pp. 559–571, Sept. 1994.

[16] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. Image Processing*, vol. 3, pp. 572–588, Sept. 1994.

[17] B.-J. Kim and W. A. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees," in *Proc. IEEE Data Compression Conf.*, Mar. 1997, pp. 251–260.

[18] S.-J. Choi and J. W. Woods, "Motion-compensated 3-D subband coding of video," *IEEE Trans. Image Processing*, vol. 8, pp. 155–167, Feb. 1999.

[19] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 301–317, Mar. 2001.

[20] NAB '98 – Lucent Demos New HDTV Encoder (1998, Apr.). [Online]. Available: http://www.bell-labs.com/news/1998/april/6/1.html

[21] NDS Demos Multiplex (1998, Nov.). [Online]. Available: http://web-star.com/hdtvnewsonline/releases/NDSDemosMultiplex.html

[22] A. Puri and A. Wong, "Spatial domain resolution scalable video coding," in *Proc. SPIE Conf. Visual Communications and Image Processing*, Boston, MA, Nov. 1993, pp. 718–729.

[23] R. Dugad and N. Ahuja, "A fast scheme for image size change in the compressed domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 461–474, Apr. 2001.

[24] (1996) MPEG-2 Video Codec (With Source Code). MPEG Software Simulation Group (MSSG). [Online]. Available: http://www.mpeg.org/MPEG/MSSG/

[25] C. A. Gonzales, H. Yeo, and C. J. Kuo, "Requirements for motion-estimation search range in MPEG-2 coded video," *IBM J. Res. Develop.*, vol. 43, no. 4, pp. 453–470, July 1999.