SPIHT Video Coder
Jayashree Karlekar and U. B. Desai
Dept. of Electrical Engineering,
Indian Institute of Technology - Bombay,
Powai, Mumbai - 400 076 INDIA
Email: ubdesai@ee.iitb.ernet.in

ABSTRACT—In this paper we propose a new video coding scheme based on discrete wavelet transform (DWT), as it provides a better way to address scalability functionalities, than MPEG-2. To code wavelet coefficients efficiently, we use set partitioning in hierarchical trees (SPIHT) and adaptive arithmetic coding algorithms. Motion compensation (MC) is done in spatial domain to remove temporal redundancy present between frames. To avoid blocking artifacts caused by block motion compensation, overlapping block motion compensation (OBMC) is done. The proposed video encoder is compared with MPEG-2.

I. INTRODUCTION

The discrete wavelet transform (DWT) has recently received a lot of attention due to the multiresolution representation it provides of a signal with localization in both time and frequency. This property is very desirable in image and video coding applications. The use of DWT to image coding application has become much more interesting after Shapiro introduced and popularized zerotree structure in [1] to code images at very high compression ratios with good quality. The embedded zerotree of wavelet coefficients (EZW) algorithm of [1] though gives better results, the set partitioning in hierarchical trees (SPIHT) [5] is recognized as one of the best methods available in literature to encode still images at target bit-rate/distortion. SPIHT gives around 2 dB improvement over the EZW for same bit-rate.

In this paper we propose a new video coding technique based on DWT which gives quality of compressed video comparable to MPEG-2 [4]. A similar method based on zerotree concept has been proposed for very low bit rate video coding in [2]. The specific building blocks of the proposed video coder are: 1) DWT to remove spatial correlation present in each frame before and after motion compensation is done, 2) block motion compensation in spatial domain to reduce temporal redundancy present between frames, 3) overlapped blocked motion compensation to avoid blocking artifacts caused by block motion compensation so that cohesive input is presented for DWT calculations, 4) SPIHT in DWT domain to quantize wavelet coefficients and 5) adaptive arithmetic coder to code output given by SPIHT. Motion compensation is done in spatial domain on block-size 16x16 with half-pel accuracy. To avoid discontinuities at boundaries of each block arising from block motion compensation and which gives rise to high frequencies in wavelet transform domain, the overlapped block motion compensation (OBMC) of H.263 [3] is used so that cohesive input is presented to DWT calculations. The proposed method provides a better way to address the scalability functionalities than MPEG-2, because of multiresolution structure inherent in DWT. The proposed algorithm is compared with MPEG-21 for ballet video sequence of size 704x480, at 30 frames per second rate.

Organization of the paper is as follows. In Section II a brief introduction of the SPIHT algorithm is given. Section III outlines the video coding and decoding algorithm. Simulation results are given in Section IV, while Section V concludes the paper.

II. SPIHT Algorithm

Embedded zerotree of wavelet coefficients (EZW) [1] algorithm is a technique for coding wavelet transform coefficients. Besides superior compression, the advantages of EZW coding include simplicity, an embedded bitstream, scalability and precise bit-rate control. EZW is based on three concepts: 1) exploiting the self-similarity inherent in the wavelet transform to predict the significant information across scales; 2) successive approximation quantization of the wavelet coefficients; and 3) lossless compression using adaptive arithmetic coding.

Set Partitioning in Hierarchical Trees (SPIHT) is in principle similar to EZW, the crucial differences being the way coefficients are partitioned and how the significant information is conveyed to the decoder. Here we briefly outline the SPIHT algorithm, for more details refer to [5].

SPIHT algorithm is applied to coefficients resulting from a DWT. Normally, most of the image energy in DWT is concentrated in the low frequency components shown with * in Fig. 1. The variance decreases from highest to the lowest (coarsest to finest) levels of subband pyramid. A tree structure, called spatial orientation tree, defines the spatial relationship on the hierarchical pyramid, similar to zerotree structure of EZW. Fig. 1 shows the spatial orientation tree defined in a pyramid constructed with recursive four-subband splitting. Each node of the tree corresponds to a pixel, and
significance is important, the significance information
presents either the set
identified by a coordinate
is stored in three ordered lists, called
between them we call the LIS entry is of type D if it
is significant then it is partitioned into
four offsprings
spatial orientation in the next finer level of the pyramid.
The tree is defined in such a way that each node has
either no offsprings (leaves) or four offsprings
shown
as
Fig. 1. The pixels in the highest level of the pyramid
are the tree roots.

The following sets of symbols are used to present the
algorithm:
- \( \mathcal{O}(i,j) \): set of offspring-nodes of \((i,j)\),
- \( \mathcal{D}(i,j) \): set of all descendants of \((i,j)\),
- \( \mathcal{L}(i,j) = \mathcal{D}(i,j) - \mathcal{O}(i,j) \),
- \( \mathcal{H} \): coordinates of all spatial orientation tree roots
(nodes in the highest pyramid level),
- \( S_n(i,j) \) is a function to indicate the significance of
a set of coordinates \((i,j)\).

Then the set partitioning rules are:
1. the initial partition is formed with the sets \((i,j)\)
and \( \mathcal{D}(i,j) \in \mathcal{H} \),
2. if \( \mathcal{D}(i,j) \) is significant then it is partitioned
into \( \mathcal{L}(i,j) \) plus the four single-element sets with
\((k,l) \in \mathcal{O}(i,j) \),
3. if \( \mathcal{L}(i,j) \) is significant then it is partitioned into
the four sets \( \mathcal{D}(k,l) \) with \((k,l) \in \mathcal{O}(i,j) \)

Since the order in which the subsets are tested for
significance is important, the significance information
is stored in three ordered lists, called list of insignifi-
cant sets (LIS), list of insignificant pixels (LIP), and
list of significant pixels (LSP). In all lists each entry is
identified by a coordinate \((i,j)\), which in the LIP and
LSP represents individual pixels, and in the LIS repres-
ts either the set \( \mathcal{D}(i,j) \) or \( \mathcal{L}(i,j) \). To differentiate
between them we call the LIS entry is of type \( D \) if it
represents \( \mathcal{D}(i,j) \), and of type \( L \) if it represents \( \mathcal{L}(i,j) \).

Algorithm:
1. Initialization:
   - output \( n = \lfloor \log_2(\max(c_{i,j}) \cdot c_{i,j}!) \rfloor \), where \( c_{i,j} \) are
   wavelet transform coefficients; set the LSP as an
   empty list, and add the coordinates \((i,j) \in \mathcal{H} \) to
   both the LIP and to the LIS, as type \( D \) entries.
2. Sorting pass:
   (a) for each entry \((i,j) \) in the LIP do:
       - if \( S_n(i,j) = 1 \) then move \((i,j) \) to the LSP
       - and output sign of \( c_{i,j} \)
   (b) for each entry \((i,j) \) in the LIS do:
      i. if the entry is of type \( D \) then output \( S_n(D(i,j)) \);
         if \( S_n(D(i,j)) = 1 \) then
         - for each \((k,l) \in \mathcal{O}(i,j) \) output \( S_n(k,l) \);
            - if \( S_n(k,l) = 1 \), add \((k,l) \) to the LSP
            - and output sign of \( c_{k,l} \);
            - if \( S_n(k,l) = 0 \), add \((k,l) \) to the end of
              the LIP;
         - if \( L(i,j) \neq \emptyset \) then move \((i,j) \) to the end
           of the LIS as an entry of type \( L \);
      ii. if the entry is of type \( L \) then output \( S_n(L(i,j)) \);
         if \( S_n(L(i,j)) = 1 \) then add each \((k,l) \in \mathcal{O}(i,j) \)
            to the end of LIS as an entry of type \( D \);
            remove \((i,j) \) from the LIS.
3. Refinement pass: for each entry \((i,j) \) in the
   LSP, except those included in the last sorting pass,
   output the \( n \)th most significant bit of \( |c_{i,j}| \).
4. Quantization-step update: decrement \( n \) by 1
   and go to step 2.

III. VIDEO ENCODER-DECODER

Here we present a brief description of the proposed
video encoder and decoder. Each frame of video se-
quence is encoded either as intra ("I") or inter ("P",
forward prediction) frame, where intra frame is used for
the first and every 15th frame while inter frame is used
for the remaining frames of the video sequence. Intra
frames are coded using SPIHT and adaptive arithmetic
coder [6]. The remaining frames of the video sequence
are encoded by forward predicted "P" frames from the
preceding reconstructed frames. SPIHT is recognized
as one of the best ways to encode still images at a tar-
get bit rate. Embedded feature of the algorithm makes
it possible to encode the frames at exactly the chosen
rate.

For inter frames, a block based motion estimation
scheme is used to detect local motion on 16x16 sized
blocks to half-pel accuracy. Each block is predicted using
overlapping block motion compensation scheme of
H.263. After all the blocks have been predicted, the
residuals are put together to form a complete residual
frame for subsequent processing by the DWT. The over-
lapping in motion compensation ensures that a coherent
residual is presented to the DWT block, without any
artificial block discontinuities. For those blocks where
motion estimation fails, the intra mode is selected and
the original image block is coded. To turn this block
Motion Estimation and Compensation:

The proposed video encoder uses the block motion estimation scheme of H.263 [3]. Motion estimation is performed on 16x16 blocks of luminance component only. The distortion measure used is the sum of absolute difference (SAD) as opposed to mean square error to make the motion estimation process fast. A full search is used and the search area is ±15 pixels in all directions from the center of the block. The SAD for the zero translation motion vector for 16x16 block is reduced by a bias, set to 100 by default, to favor the zero motion vectors.

The motion estimation scheme also has inter/intra mode decision. For each 16x16 block, its mean value is also calculated. The SAD between the block and its mean is also calculated which is called MSAD. If this value (MSAD) is smaller than the SAD calculated by motion estimation by a set margin, 500 by default, the intra mode is chosen and no motion vectors are sent. In this case block is predicted by its mean and the mean is sent as overhead. Otherwise, the inter mode is chosen and the block is estimated using motion vectors.

In order to achieve better estimates, the half-pel motion estimation is done using previous reconstructed frames. The search is performed only on the luminance component. The range of search is one-half pixel in all four directions. Bilinear interpolation is used to obtain half pixels. For the chrominance components, motion vectors are divided by two and also quarter-pel interpolation is done to obtain the predictions of the chroma blocks.

Overlapping block motion compensation (OBMC) is an advanced scheme for block motion compensation that overlaps neighboring blocks prior to subtraction from the current block in order to reduce the effect of block boundary discontinuities. This overlapping provided a coherent motion-compensated frame free of artificial block discontinuities that a non-overlapping motion compensation scheme would produce. OBMC is performed for luminance as well as chrominance components also.

Rate Control:

Embedded nature of the SPIHT algorithm gives precise control over the rate/distortion for each frame of the video sequence. But keeping the rate constant for each frame in a video sequence does not guarantee a constant or near-constant level of distortion which is more desirable than one with lower average distortion but higher variability. This is because human viewers tend to find frequent changes in quality more noticeable and annoying. Incorporating the distortion criteria in SPIHT algorithm slows down the encoder speed. So, to give near-constant quality video sequence without increasing the computational complexity, we propose here to use the threshold as a criterion for rate control in SPIHT algorithm. This threshold is equivalent to the quantization parameter used in H.263, MPEG-1 or MPEG-2. Keeping the threshold constant over the video sequence gives near-constant quality output.

Algorithm:

- For Intra frame:
  - take discrete wavelet transform of the frame.
  - quantize and code wavelet coefficients using SPIHT and adaptive arithmetic coder till target bit-rate/distortion is achieved.

- For Inter frames:
  - use of block motion compensation to track local motion. The previous reconstructed frame is used for motion compensation.
  - for the blocks where prediction fails, the intra block mode is selected. The residual of such blocks is formed by subtracting the mean from the block and sent as overhead
  - use of overlapping block motion compensation to reduce blocking artifacts so that coherent residual is presented to the wavelet transform
  - take discrete wavelet transform of the residual image obtained by block prediction
  - quantize and code wavelet coefficients using SPIHT and adaptive arithmetic coder till target bit-rate/distortion is achieved.

The block diagram of the proposed video encoder scheme is shown in Figure 2. The video decoder is exactly reverse of the video encoder.
IV. EXPERIMENTAL RESULTS

The proposed video coder has been run to encode I and P frames using block motion estimation of sizes 16x16, overlapping motion compensation, the discrete wavelet transform using orthogonal filter of Daubechies' with regularity four. Each frame is decomposed into five levels of wavelet decomposition. As video sequence is in 4:2:0 format, only four level wavelet decomposition is taken for chrominance components.

We have carried out experiments on video sequences obtained from the net. In particular, we present results for the ballet video sequence, of size 704x480 and for first 90 frames. The PSNRs (peak signal to noise ratio) for the proposed video coder and MPEG-2 are displayed in Figure 3. We find that the proposed scheme gives comparable results as compared to MPEG-2 at rate around 3.5 Mega bits per second (MBPS) at 30 frames per second (FPS). The encoder of the proposed scheme is somewhat faster, while decoder is comparable to that of MPEG-2.

V. CONCLUSION

The proposed video coding algorithm gives comparable results to MPEG-2 at improved coding efficiency. The algorithm provides a better way to address the scalability functionalities of MPEG-2 because of inherent multiresolution structure present in wavelet transform.

The proposed video encoder can be improved upon by incorporating "B" (bidirectional prediction) frames and with better rate control scheme.

REFERENCES


Fig. 3. PSNR results for the ballet video sequence. Dotted lines shows the values for MPEG-2 and solid lines for the proposed algorithm. Top most plot (a), shows the results for the luminance and remaining two (b) and (c), shows for the chrominance components. The results are shown for 3.5 MBPS bit rate.