Bidirectional MC-EZBC With Lifting Implementation

Peisong Chen and John W. Woods
chen@cipr.rpi.edu woods@ecse.rpi.edu

Center for Next Generation Video
Rensselaer Polytechnic Institute
Troy, NY 12180-3590, USA

This material is partially based upon work supported by the National Science Foundation IUCRC Program under Grant No. EEC-9812706. The government has certain rights in this material.

Abstract

In conventional motion compensated 3-D subband/wavelet coding, where the motion compensation is unidirectional, incorrect classification of connected and unconnected pixels caused by incorrect motion vectors (MVs) has resulted in some coding inefficiency and visual artifacts in the embedded low frame-rate video. In this paper, we introduce bidirectional motion compensated temporal filtering (MCTF) with unconnected pixel detection and *I* blocks. We also incorporate a recently suggested lifting implementation of the subband/wavelet filter for improved MV accuracy in an MC-EZBC coder. Simulation results compare PSNR performance of this new version of MC-EZBC versus H.26L under the constraint of equal GOP size, and show a general parity with this state-of-the-art nonscalable coder on several test clips.

I. Introduction

In MPEG hybrid coding, temporal redundancy is removed by motion-compensated prediction (MCP). A video is typically divided into a series of groups of pictures (GOP), where each GOP begins with an intra-coded frame (I) followed by an arrangement of forward predictive-coded frames (P) and bidirectional predicted frames (B). Both P-frames and B-frames are interframes [10]. Bidirectional prediction, also called motion-compensated (MC) interpolation, is a key feature of MPEG video. B-frames coded with bidirectional prediction generally use two reference frames, one in the past and one in the future. A target macroblock in a B-frame can be predicted from the past reference frame (forward prediction) or from the future reference frame (backward prediction), or by an average of two prediction macroblocks, one from each reference frame (interpolation) [10]. The respective blocks are called P-block, B-block, and I-block, respectively.

Motion-compensated 3-D subband coding (MC-3DSBC) [3], [15] removes temporal redundancy by motion-compensated temporal filtering (MCTF). Such coders do not suffer the drift problem often exhibited by scalable hybrid coders with their incorporated feedback loops. In [9], the authors presented a completely non-hybrid video coding system MC-EZBC, a family of subband/wavelet coders that exploit temporal correlation but are fully embedded in quality/bit-rate, spatial resolution, and frame rate. In this system, an invertible motion compensated 3-D subband/wavelet filter bank was utilized for video analysis/synthesis. The efficient embedded image coding scheme EZBC [8] was extended to code the video subbands, but all motion estimation was unidirectional.

MCTF plays an essential role in MC-EZBC to exploit temporal redundancy in image sequences. A challenging problem in MCTF is to realize sub-pixel accurate motion compensation. Sub-pixel accurate motion compensation is especially useful for low spatial resolution video, where the sample rate is near to or below the Nyquist limit and the power spectrum is flatter. In hybrid coders such as MPEG-2 [10], the use of half-pixel accurate motion compensation has achieved substantial coding gain with respect to the use of pixel accuracy. In the very efficient H.26L [6], even $\frac{1}{8}$-pixel accurate motion compensation has been found useful. In hybrid coders, the reconstructed frames are used as reference frames for motion compensated prediction and interpolation. Any sub-pixel accurate spatial interpolation can be duplicated at the decoder, and thus achieve *invertibility*, i.e. perfect reconstruction in the absence of quantization errors. Combining sub-pixel motion vector (MV) accuracy with MCTF though, while maintaining the invertibility property, has proved more elusive. Hsiang and Woods [7] designed an invertible MCTF with half-pixel accuracy by incorporating the spatial interpolation as part of the subband/wavelet filtering. Significant coding gain was observed on test SIF sequences. While this scheme achieved perfect reconstruction, it was not readily extensible to higher motion compensation accuracy. Recently, the lifting implementation of subband/wavelet filters has been used for MC temporal subband decomposition [17] [19] to provide invertibility for arbitrary sub-pixel accuracy. In all three schemes, the required spatial interpolation is incorporated into the perfect reconstruction subband/wavelet temporal decomposition, effectively making the MC temporal filtering slightly spatial also.

Another challenging problem in MCTF is that filtering across poorly matched pixels not only decreases coding efficiency but also can create so-called *motion artifacts* in the low frame-rate video. In this paper, we detect these poorly matched pixels and introduce a new bidirectional MCTF front end for MC-EZBC. After a review of MCTF, we discuss our lifting implementation of sub-pixel accurate MCTF. We then discuss unconnected and poorly matched connected pixels and their influence on artifacts in the embedded low frame-rate video as well as on coding efficiency. By using a pixel-based detection algorithm, we locate the real unconnected pixels. Poorly matched connected pixels are also singled out. Since we use a block-based motion field, the positions of these pixels will be conveyed in a block-based way. Experimental results then follow, including a PSNR performance comparison with H.26L.

II. Motion-Compensated Temporal Filtering (MCTF) Review

The quality of the MCTF plays an essential role in motion compensated 3-D non-recursive subband/wavelet coding. It will influence not only the coding efficiency, but also the quality of the resulting low frame-rate video. In the following, we review two approaches for MCTF.

A. Noninvertible Approach

This processor was proposed by Ohm [15] and extended by Choi and Woods [3]. We term it *noninvertible approach* only because it is not invertible at sub-pixel MCTF, while it is invertible at integer-pixel accuracy. We show a 2-tap MCTF on two frames in Fig. 1, where only integer motion displacement is considered. The arrows indicate forward motion estimation with the first frame $(2t)$ in a frame pair used as reference for the second $(2t+1)$, where t is an integer. Since we use block matching motion estimation, the blocks are delineated by horizontal lines in the figure. We can see the pixels in Fig. 1 classified as *connected* and *unconnected*. If there is a one-to-one connection between the pixels, they are connected pixels. If several pixels in frame two connect to the same pixel in frame one, only one of them is classified as a connected pixel, the others are listed as unconnected. A scan-order rule is used to discriminate. Conversely, unconnected pixels in frame one are not used as reference for frame two. We can see that the total number of unconnected pixels is the same in both frames. After the classification, we do the actual motion compensated temporal filtering only on the connected pixels. The unconnected pixels are not

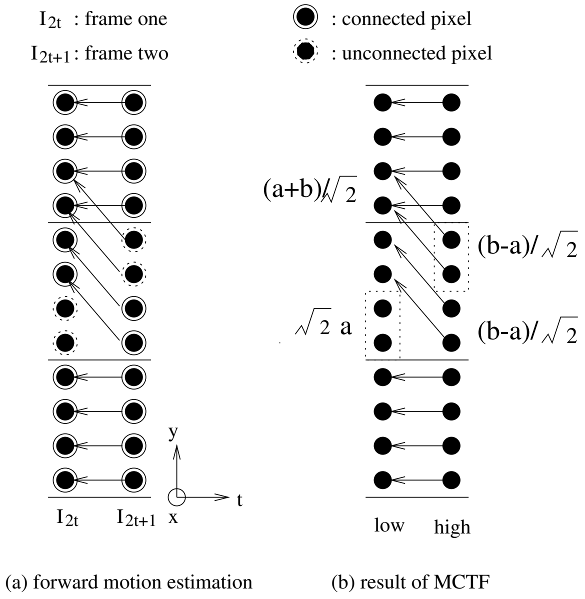


Fig. 1. MCTF scheme for a given frame pair.

filtered. The result is shown in Fig. 1(b), where “a” represents the value of a pixel in frame one, and “b” represents the value of a pixel in frame two.

When the MV has an integer value, the subband analysis pair for connected pixels is

$$L[m-d_m,n-d_n]=\frac{1}{\sqrt{2}}I_{2t+1}[m,n]+\frac{1}{\sqrt{2}}I_{2t}[m-d_m,n-d_n] \tag{1}$$

$$H[m,n]=\frac{1}{\sqrt{2}}I_{2t+1}[m,n]-\frac{1}{\sqrt{2}}I_{2t}[m-d_m,n-d_n], \tag{2}$$

where $L[m,n]$ and $H[m,n]$ are the temporal low and high frequency frames, $I_{2t}[m,n]$ and $I_{2t+1}[m,n]$ are the first and second frames, and (d_m,d_n) is the motion vector. The orthonormal basis functions $[\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}]$ for the lowpass filter and $[\frac{1}{\sqrt{2}},\frac{-1}{\sqrt{2}}]$ for the highpass filter are employed. They constitute the impulse responses of a perfect reconstruction QMF pair.

For the unconnected pixels in frame one, their scaled original values are inserted into the temporal low subband,

$$L[m,n]=\frac{2I_{2t}[m,n]}{\sqrt{2}}. \tag{3}$$

For the unconnected pixels in frame two, the scaled displaced frame differences (DFD) are substituted into the temporal high subband [3]. Fig. 1(b) shows that the temporal low subband is time referenced to that of frame one, while the temporal high subband is time referenced to that of frame two. Note there is no test of these matches, other than they are the best found, and hence, every pixel in frame two is said to possess an MV.

1) Sub-pixel Accuracy: In sub-pixel accurate MCTF, we have to first define connection in the case of sub-pixel MVs. When a displacement (d_m,d_n) from frame two points to a sub-pixel position in frame one as shown in Fig. 2, we can say $I_{2t+1}[m,n]$ is connected to $I_{2t}[m-\bar{d}_m,n-\bar{d}_n]$, where $\bar{d}_m$ and $\bar{d}_n$ are defined as follows. If d_m points to a half-pixel position, $\bar{d}_m = \lfloor d_m \rfloor$ where $\lfloor \cdot \rfloor$ denotes the least integer function; in the other cases, $\bar{d}_m$ represent the closest integer values to d_m. $\bar{d}_n$ is defined in the same way. The highpass coefficient comes from the filtering of $I_{2t+1}[m,n]$ and the interpolated reference pixel $\tilde{I}_{2t}[m-d_m, n-d_n]$.

$$H[m,n] = \frac{1}{\sqrt{2}} I_{2t+1}[m,n] - \frac{1}{\sqrt{2}} \tilde{I}_{2t}[m-d_m, n-d_n]. \tag{4}$$

Since we want lowpass coefficients to be at integer pixel positions, we do the lowpass filtering of $I_{2t+1}[m,n]$'s connected integer pixel $I_{2t}[m-\bar{d}_m, n-\bar{d}_n]$ and the interpolated pixel $\tilde{I}_{2t+1}[m-\bar{d}_m+d_m, n-\bar{d}_n+d_n]$, as

$$L[m-\bar{d}_m, n-\bar{d}_n] = \frac{1}{\sqrt{2}} \tilde{I}_{2t+1}[m-\bar{d}_m+d_m, n-\bar{d}_n+d_n] + \frac{1}{\sqrt{2}} I_{2t}[m-\bar{d}_m, n-\bar{d}_n]. \tag{5}$$

Here we use the inverse of $I_{2t+1}[m,n]$'s forward MV as $I_{2t}[m-\bar{d}_m, n-\bar{d}_n]$'s backward MV. In general, this scheme is not invertible [3]. Since we can only use H and L to reconstruct I_{2t}, but H contains the information on interpolated pixels in I_{2t}, if this interpolation itself is not invertible, we cannot reconstruct I_{2t} exactly. Unconnected pixels in I_{2t+1} are processed like (4), and unconnected (unreferred) pixels in I_{2t} are processed as in (3).

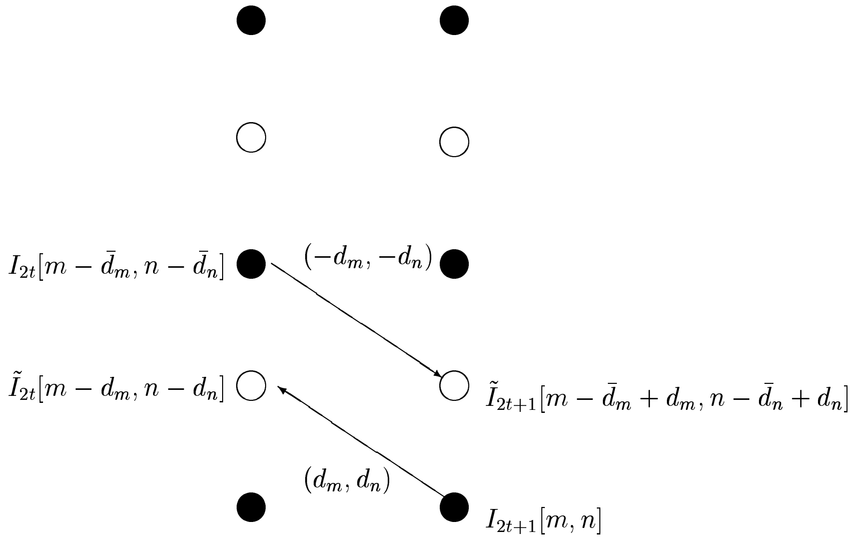

$I_{2t}[m-\bar{d}_m, n-\bar{d}_n]$
$(-d_m, -d_n)$
$\tilde{I}_{2t}[m-d_m, n-d_n]$
$\tilde{I}_{2t+1}[m-\bar{d}_m+d_m, n-\bar{d}_n+d_n]$
(d_m, d_n)
$I_{2t+1}[m,n]$

Fig. 2. Sub-pixel accurate MCTF (Choi and Woods' scheme).

B. Invertible Half-pixel Accurate MCTF

Hsiang and Woods [7] designed an invertible half-pixel accurate MCTF. In this approach, when a motion displacement (d_m, d_n) from frame two points to a half-pixel position in frame one, a composite block is constructed by merging the pair of linked motion blocks. Then the composite block is decomposed by a spatial two-channel subband/wavelet analysis filter bank, with the lowpass output and the highpass output being put into the temporal low subband and the temporal high subband respectively. So perfect reconstruction can be realized, by the assumed invertibility of the chosen spatial analysis/synthesis pair. Effectively the desired spatial interpolation has been incorporated into the subband/wavelet temporal filter, and has therefore become invertible.

III. Sub-pixel Accurate MCTF Using a Lifting Implementation

Since MCTF is a subband/wavelet transform, we can implement it using a lifting scheme [1]. The parallel lowpass and highpass filtering branches are then converted to a serial computation. For sub-pixel accurate MCTF, the lifting implementation can guarantee invertibility. We only use Haar filters here and in the experiments at the end, but the results are extendible to longer filters directly.

A. Lifting Implementation

The so-called lifting scheme is a recent generalization of subband/wavelet transforms [1]. Lifting was introduced into sub-pixel MCTF [17] to allow perfect reconstruction, independent of the interpolation method used. Here, we take a look at this implementation for connected pixels. If the MVs have sub-pixel accuracy, the lifting scheme still calculates the temporal highpass frame in the same way as [3],

$$H[m,n] = \frac{1}{\sqrt{2}} I_{2t+1}[m,n] - \frac{1}{\sqrt{2}} \tilde{I}_{2t}[m-d_m, n-d_n]. \tag{6}$$

For the lowpass frame, there are two proposed ways to get needed backward motion vectors. One way [19] uses backward motion estimation, thereby requiring two motion fields to be coded. The other way, followed in [3] and [17], requires only one motion field and is chosen for use here. The inverse of $I_{2t+1}[m,n]$'s forward MV is used as $I_{2t}[m-\bar{d}_m, n-\bar{d}_n]$'s backward MV, so we obtain

$$L[m-\bar{d}_m, n-\bar{d}_n] = \tilde{H}[m-\bar{d}_m+d_m, n-\bar{d}_n+d_n] + \sqrt{2} I_{2t}[m-\bar{d}_m, n-\bar{d}_n]. \tag{7}$$

At the decoder, by using L and H, we can perform an identical interpolation on H and reconstruct I_{2t} exactly if there is no quantization error,

$$I_{2t}[m-\bar{d}_m, n-\bar{d}_n] = \frac{1}{\sqrt{2}} L[m-\bar{d}_m, n-\bar{d}_n] - \frac{1}{\sqrt{2}} \tilde{H}[m-\bar{d}_m+d_m, n-\bar{d}_n+d_n]. \tag{8}$$

After I_{2t} is available, we can also perform an identical interpolation on I_{2t} as at the encoder, and reconstruct I_{2t+1} exactly as

$$I_{2t+1}[m,n] = \sqrt{2} H[m,n] + \tilde{I}_{2t}[m-d_m, n-d_n]. \tag{9}$$

So no matter how we interpolate these subpixels, if we do it in the same way at the encoder and decoder, we can achieve perfect reconstruction. Again, this occurs because we have incorporated the desired spatial interpolation into the guaranteed reconstructible lifting filters. In (8), we see L and H are still necessary for the reconstruction of I_{2t}, and H only contains the information of interpolated pixels in I_{2t}. But this identical interpolation is present in L, so it is canceled out in (8). Thus the interpolation algorithm has no influence on the MCTF invertibility. Of course, there is still the question of which interpolation is best to use. Unconnected pixels in I_{2t+1} are processed like (4), and unconnected (unreferred) pixels in I_{2t} are processed as in (3).

B. Sub-pixel Interpolation

Since the interpolated temporal highpass band $\tilde{H}$ is used to update I_{2t}, the question arises: Can we make $L[m - \bar{d}_m, n - \bar{d}_n]$ the same as (5), since that should be close to optimal? To answer this question, we use I_{2t} and I_{2t+1} to represent L as

$$\begin{aligned} L[m-\bar{d}_m, n-\bar{d}_n] &= \tilde{H}[m-\bar{d}_m+d_m, n-\bar{d}_n+d_n] + \sqrt{2}I_{2t}[m-\bar{d}_m, n-\bar{d}_n] \\ &= (\tilde{I}_{2t+1}[m-\bar{d}_m+d_m, n-\bar{d}_n+d_n] - \tilde{\tilde{I}}_{2t}[m-\bar{d}_m, n-\bar{d}_n])/\sqrt{2} \\ &\quad +\sqrt{2}I_{2t}[m-\bar{d}_m, n-\bar{d}_n] \\ &= \frac{1}{\sqrt{2}}(2I_{2t}[m-\bar{d}_m, n-\bar{d}_n] - \tilde{\tilde{I}}_{2t}[m-\bar{d}_m, n-\bar{d}_n]) \\ &\quad +\frac{1}{\sqrt{2}}\tilde{I}_{2t+1}[m-\bar{d}_m+d_m, n-\bar{d}_n+d_n]. \end{aligned} \tag{10}$$

The term $\tilde{\tilde{I}}_{2t}[m - \bar{d}_m, n - \bar{d}_n]$ is the result of two successive interpolations: the first using integer pixels in I_{2t} to interpolate subpixels as in (6) with the information stored in the integer positions of H; the second interpolation happens in (7) when we use integer pixels in H (subpixels of I_{2t} in effect) to interpolate subpixels in H (integer pixels of I_{2t} in effect). If we can satisfy $I_{2t}[m - \bar{d}_m, n - \bar{d}_n] = \tilde{\tilde{I}}_{2t}[m - \bar{d}_m, n - \bar{d}_n]$, then

$$L[m-\bar{d}_m, n-\bar{d}_n] = \frac{1}{\sqrt{2}}\tilde{I}_{2t+1}[m-\bar{d}_m+d_m, n-\bar{d}_n+d_n] + \frac{1}{\sqrt{2}}I_{2t}[m-\bar{d}_m, n-\bar{d}_n]. \tag{11}$$

This is the same as (5). Since the pixels in $\tilde{\tilde{I}}_{2t}[m-\bar{d}_m, n-\bar{d}_n]$ undergo interpolation from integer pixels to subpixels and then from subpixels to integer pixels, the necessary conditions are 2-D separable sinc function interpolation and a constant MV over the support region of the interpolation filter. Considering the component 1-D filters, for various sub-pixel positions s, we have ideal interpolation filters

$$f(n+s) = \sum_m f(m)\frac{sin\pi(n+s-m)}{\pi(n+s-m)}, 0 < s < 1. \tag{12}$$

For application, we employ FIR interpolation filters using the Hamming window [18], with filter coefficients given in Table I. We implement the separable interpolation first in the column direction and then in the row direction.

IV. Motion Estimation with Mismatched Block Detection

In motion estimation, we find the MV for each pixel of frame two. Several pixels may choose the same reference pixel. There are good matches and bad matches based on their displaced frame difference (DFD). If we have

$\frac{1}{8}$-pixel	$\frac{1}{4}$-pixel	$\frac{3}{8}$-pixel	$\frac{1}{2}$-pixel	$\frac{5}{8}$-pixel	$\frac{3}{4}$-pixel	$\frac{7}{8}$-pixel
-0.0072	-0.0110	-0.0117	-0.0105	-0.0081	-0.0053	-0.0026
0.0284	0.0452	0.0505	0.0465	0.0363	0.0233	0.0105
-0.0902	-0.1437	-0.1624	-0.1525	-0.1224	-0.0812	-0.0380
0.9742	0.8950	0.7713	0.6165	0.4465	0.2777	0.1249
0.1249	0.2777	0.4465	0.6165	0.7713	0.8950	0.9742
-0.0380	-0.0812	-0.1224	-0.1525	-0.1624	-0.1437	-0.0902
0.0105	0.0233	0.0363	0.0465	0.0505	0.0452	0.0284
-0.0026	-0.0053	-0.0081	-0.0105	-0.0117	-0.0110	-0.0072

TABLE I

INTERPOLATION FILTER COEFFICIENTS FOR $\frac{1}{8}$-PIXEL LOCATIONS.

temporal filtering between bad matched pixels, not only do we get a DFD with high energy, but also we get temporal low frequency subbands with bad visual quality.

In a temporal scalable coder using MCTF, the most natural choice for the low frame-rate data is the MCTF temporal low frequency subband output. Therefore, as mentioned above, it is important that it be nearly artifact free and visually pleasing. Note that this is in contrast with the usual choice for the low frame-rate data in the case of hybrid coders, i.e. the sub-sampled frames themselves. The MCTF should therefore lead to reduced aliasing. So the success of MCTF depends on energy being concentrated in the temporal low subband and visually appealing low frame-rate video.

In conventional MCTF methods [3] [15], the classification of pixels as unconnected depends on the scan order. Usually no test of MV accuracy is used, so that the motion field is defined for every MV block in frame two, even though some of these blocks may not really have a good match in frame one. These poor or simply wrong connections lead to faulty classification of connected and unconnected pixels. Hence, both the motion estimation and the decision of unconnected pixels are questionable. To have visually appealing low frame-rate data, these poorly matched pixels should be first singled out and then processed as unconnected pixels.

In this section, based on information obtained from the forward motion estimation, we design a pixel-based algorithm to detect true unconnected pixels. We also single out connected pixels with poor match.

A. HVSBM

We use hierarchical variable size block matching (HVSBM) for motion estimation, to speed up the search process and impose smoothness between neighboring MVs. Fig. 3 shows the basic idea of 3-level HVSBM. The last level of the pyramid reveals the quadtree structure of the MVs. We start motion estimation from the coarsest level of the pyramid. Our matching criterion is the mean absolute difference (MAD). At the next level, we first refine all MVs found in the upper level in a small refinement region, then each block on the quadtree is subdivided into four

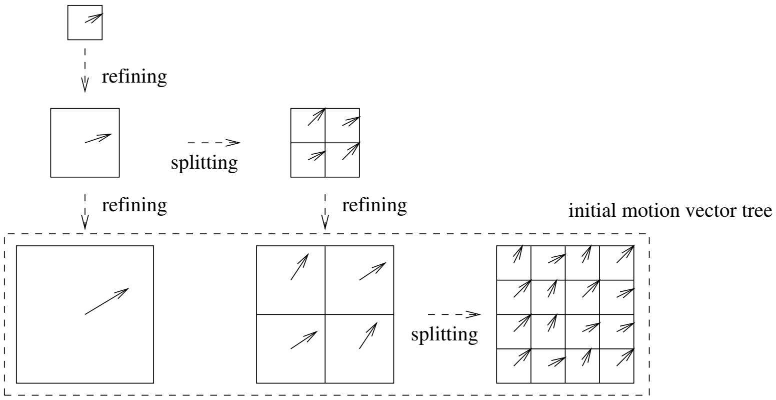


Fig. 3. A 3 level HVSBM showing 3 subband levels.

subblocks. The MVs for new born subblocks are also generated by refining the MV of their parent in the upper level. This process continues till the last level of the pyramid. The refinement region is always $[-1.5,\ 1.5] \times [-1.5,\ 1.5]$ with half-pixel accuracy except at the last pyramid level, where it starts from $[-1,\ 1] \times [-1,\ 1]$ for integer-pixel accuracy, and goes on to half-pixel, quarter-pixel, or eighth-pixel accuracy as required. Consequently, long-range smoothness is enforced at lower resolution (higher scale) levels, while short-range consistency is enforced at higher resolution (lower scale) levels. After generating the full MV quadtree, we first detect mismatched blocks (to be discussed below). Then a bottom-up merge [3] is employed to realize a variable block-size motion field. In sub-pixel accurate motion estimation, we use the same interpolation filters, as in Section III-B for motion compensation.

B. Detection of Unconnected Blocks

To avoid filtering across poorly matched pixels, we introduce a pixel-based algorithm to find the true unconnected pixels in frame two as illustrated in Fig. 4.

In order to find the true unconnected pixels, we must have a detection phase. In the following, we give a pixel-based algorithm to find the true unconnected pixels, as in Fig. 4, which illustrates integral motion displacements. There are four steps in this algorithm:

step 1. Do forward motion estimation in frame two.

step 2. Get the state of connection of each pixel in frame one. We define three states:

unreferred: a pixel which is not used as reference.

uni-connected: a pixel which is used as reference by only one pixel in frame two.

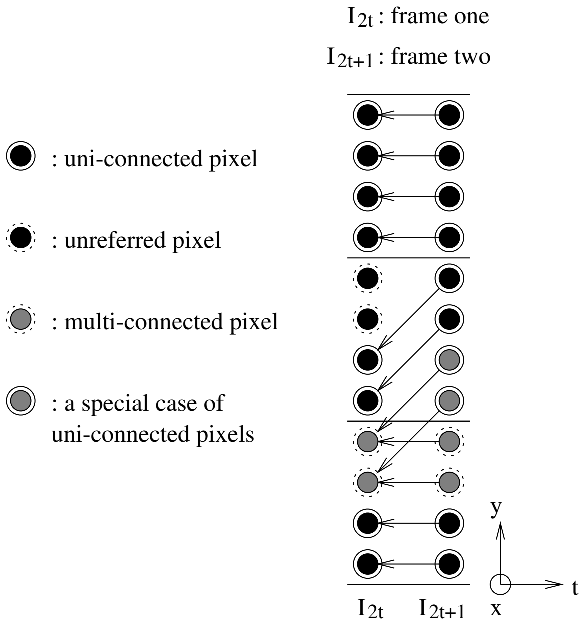
multi-connected: a pixel which is used as reference by more than one pixel in frame two.

A multi-connected pixel in frame one has several corresponding pixels in frame two, so we compute the absolute DFD value with each of them, and retain only the one with minimum value.

step 3. Get the state of connection of each pixel in frame two. There are just two states:

uni-connected: Here we have three cases

case 1: a pixel whose reference in frame one is uni-connected.



forward motion estimation

Fig. 4. State of connection of each pixel.

case 2: a pixel whose reference in frame one is multi-connected, except if its absolute DFD value with the reference pixel is the only minimum, we declare it uni-connected.

case 3: if there are several pixels in frame two pointing to the same reference pixel, and having the same minimum absolute DFD value, we settle this tie using the scan order.

multi-connected: remaining pixels in frame two.

step 4. If more than half of the pixels in a block of frame two are multi-connected, we call this block an *unconnected block* and pixels in this block are said to be *unconnected pixels*.

This algorithm can also work for MVs with sub-pixel accuracy based on our definition of connection for the sub-pixel accurate MV case. We use the interpolated subpixel to calculate the DFD.

Thus far, we have re-examined the multi-connected pixels in our MCTF, to get a near maximum number of good motion paths and to detect true unconnected pixels. Now we attempt to find the poorly matched connected pixels according to the following criterion: Calculate the mean-squared DFD and the variances of two matched blocks in frame one and frame two, excluding the already found unconnected blocks. If this mean-square DFD is larger than half of the minimum of the two variances, we reclassify the block in frame two as truly unconnected, i.e. an unconnected block. After segmenting out all of these unconnected blocks, bidirectional MCTF can be introduced naturally in the next section and we can expect better visual quality on the low frame-rate video.

V. Bidirectional MCTF with Sub-pixel Accuracy

Ohm introduced the idea of bidirectional MCTF using Haar filters in [15], where the temporal high subband was time-referenced to the first frame in a frame pair, but the motion estimation direction was backward. The unreferred pixels in the first frame were then reasonably looked at as *unconnected pixels*, but the determination of the unconnected pixels in the second frame was done according to the scan order. So this scheme still has the problem of "accidentally" classifying pixels as unconnected pixels. For the unconnected pixels in the first frame, forward MCP was utilized. DFDs with the first frame's immediately preceding reconstructed frame were substituted. These unconnected pixels were assumed to have the same motion as their neighbors, termed the "homogeneous motion" assumption in [15], so the displacement at the adjacent connected pixels were used. This might cause some problems, since there are most likely different motions around the unconnected pixels.

Our algorithm presented in Section IV-B directly addresses these problems. The pixels inside unconnected blocks will be processed as unconnected pixels.

A. Bidirectional MCTF

For unconnected blocks, we could use forward or backward motion compensated prediction, or encode them using spatial linear interpolation, and then put the interpolation error into the temporal high subband frame. We will call these *P blocks* and *I blocks* respectively. To decide which method should be used, we use linear interpolation from the neighboring four blocks and compare the sum of the absolute differences of the spatial interpolation and the sum of the absolute DFDs of the forward and backward motion compensated predictions and choose the smaller. If an unconnected block is at the right GOP boundary, only forward motion compensated prediction and spatial interpolated prediction are tested.

If backward motion estimation has the smallest DFD, the DFDs with frame two's next frame $(2t + 2)$ will be substituted as

$$H[m,n] = \frac{1}{\sqrt{2}} I_{2t+1}[m,n] - \frac{1}{\sqrt{2}} \tilde{I}_{2t+2}[m - d_m, n - d_n], \tag{13}$$

where (d_m, d_n) is the backward MV.

If forward motion estimation is still the best choice, we will use frame one as reference

$$H[m,n] = \frac{1}{\sqrt{2}} I_{2t+1}[m,n] - \frac{1}{\sqrt{2}} \tilde{I}_{2t}[m - d_m, n - d_n], \tag{14}$$

where (d_m, d_n) is the forward MV. In both these two cases, we only use the prediction stage of the lifting implementation, and skip the update stage. Otherwise linear spatial interpolation is used, and the interpolation error is put in the temporal high subband frame. Blocks choosing spatial interpolation are processed after the other blocks in a frame. We process them based on scanning order and only processed blocks can be used for the spatial interpolation.

In fact, there is another approach to process these unconnected blocks, i.e. combined forward-backward interpolation. However, in our experiments, we only observed marginal improvement with this extra computational complexity, so it is not included in the system here.

We thus have four kinds of blocks in frame two: connected blocks, P blocks using frame one ($2t$) as reference, P blocks using frame three ($2t+2$) as reference, and I blocks. We use a four-symbol Huffman code 0, 10, 110 and 111 to represent these four cases respectively, and transmit this overhead information along with the MVs. On average, this information will contribute 5 to 10 percent of the motion field bitstream.

For the lifting implementation of MCTF across connected pixels, (6) and (7) are used. For unconnected pixels in frame one whose positions can be derived from the transmitted motion field information, their scaled original values are inserted into the temporal low subband just as in (3).

B. Adaptive GOP Size

The effectiveness of motion compensation is our major concern in deciding whether to perform MC filtering. The MCTF is performed in temporal levels, starting with frame-pairs at the highest frame rate. Such filtering makes sense only when the motion information is reliable at that level. The variable size block matching motion estimation used in MC-EZBC is based on the assumption of rigid motion, which is not always valid. Furthermore, some video editing effects such as fade-in, fade-out, and dissolve will make our currently used motion estimation and compensation algorithms fail by introducing multiple local motions. In these cases, adaptation of the GOP size is useful.

Based on the percentage of unconnected pixels at the present temporal level, we decide whether to proceed with MC filtering to the next temporal level, i.e. the next lower frame rate. We find a constant threshold value of 0.5 works well over the range of input video tested. In this way, an adaptive GOP size structure is realized with a varying number of temporal decomposition levels.

C. Comparison of Bidirectional MCTF and Hybrid MPEG

We can see that the second frame in an MCTF frame pair as shown in Fig. 4 looks like the B frame in hybrid MPEG. Actually, in each level of bidirectional MCTF, the temporal high frequency frames are similar to MPEG B frames. Both the temporal pyramid structure of a multi-stage bidirectional MCTF and B frames can provide temporal scalability. Both t-H and B frames use bidirectional motion estimation. But beyond this similarity, there is a big difference. In an MPEG hybrid coder, both *reconstructed* I and P frames are used as references. The operation (temporal *DPCM*) on B frames will not influence the I and P frames. We just need to choose a motion estimation direction with low MCP error. So we do not need to know the exact positions of the unconnected blocks. But in bidirectional MCTF, we want to do temporal subband analysis on the vast connected majority of the pixels. So it is necessary to well locate unconnected blocks. Furthermore, the B frames in MPEG provide a greater degree of compression than only using I and P frames partly due to the fact that they are not used as reference for encoding other frames, so they can be quantized more coarsely, and the quantization noise does not propagate further [10] [12]. However, quantization noise in the t-H frames in MCTF leads to increased error in temporal regions of various lengths, depending on the temporal level of the t-H frame.

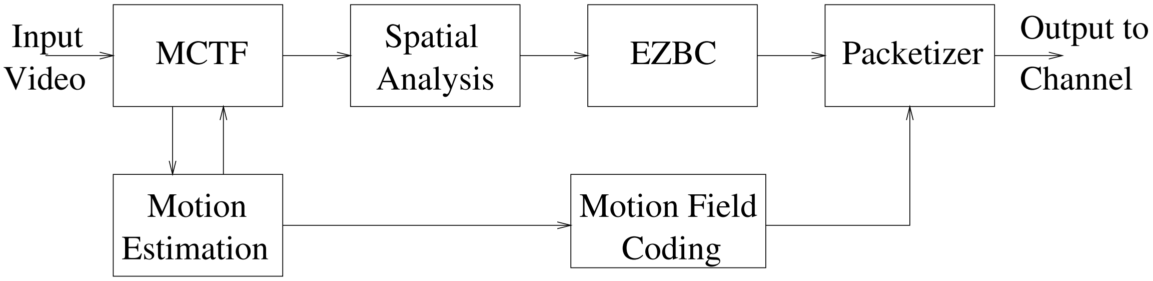


Fig. 5. Basic structure of MC-EZBC.

VI. MC-EZBC Algorithm Summary

Fig. 5 shows the basic structure of our MC-EZBC coder. The complete algorithm can be summarized as follows:

step 1: (for GOP=16) Read in 16 frames at one time.

step 2: MCTF

step 2. 1: The motion estimation for the first level MCTF is performed on the original image frames. First get forward MVs. Then find the unconnected blocks in the second frame using the algorithm of Section IV-B. Variable block size motion estimation is finalized by optimal tree pruning [3]. Based on the motion field, if the percentage of unconnected pixels of every two consecutive frames is less than 50%, a stage of temporal decomposition will be done. Otherwise, there is no temporal filtering.

step 2. 2: This procedure is then recursively performed on the generated low temporal subband. If the percentage of unconnected pixels never exceeds the threshold during this process inside a GOP, for the GOP=16 case we end up with an octave based four-level temporal decomposition, resulting in 1 t-LLLL frame, 1 t-LLLH frame, 2 t-LLH frames, 4 t-LH frames, and 8 t-H frames.

step 3: Four-stage spatial subband/wavelet analysis follows this temporal stage to complete the 3-D subband decomposition. We use Daubechies' 9/7 based spatial filters [5].

step 4: We call EZBC [8] to encode spatiotemporal subbands. EZBC is a bitplane coder. It begins with quadtree representations of individual subbands. The value of each quadtree node is just equal to the maximum magnitude of the subband coefficients in its corresponding block region. In contrast with the conventional pixel-wise bitplane coding algorithm, EZBC also needs to deal with bitplanes of nodes at individual quadtree levels. EZBC utilizes this quadtree-based zeroblock coding approach for hierarchical set-partition of subband coefficients to exploit the strong statistical dependency in the quadtree representation of the decomposed image. To code the significance of the quadtree nodes, context-based arithmetic coding is used. The context includes 8 neighbor nodes of the same quadtree level and the node of the parent subband at the next lower quadtree level.

step 5: Encode motion fields by using lossless DPCM and adaptive arithmetic coding.

step 6: In bitplane scanning to form the final bitstream, we interleave the spatial subbands of all the temporal subband frames within a GOP and further interleave their fractional coding passes [2]. To insure near constant quality across the GOPs, we stop the bitplane scanning of all the GOPs at the same fractional bit plane [2]. The bits for motion fields at the various temporal levels, are sent as overhead.

VII. Experimental Results

We have performed many experiments to demonstrate the advantages of bidirectional MCTF with lifting implementation in the framework of MC-EZBC, and present a representative set of these results here. We first compare bidirectional MCTF with unidirectional MCTF, and then show the effect of MV precision. We test the following four aspects of bidirectional MCTF, which are closely related to scalable video coding:

- motion field,
- visual quality of temporal low frequency frames,
- coding efficiency,
- temporal scalability.

The well-known test sequences *Flower Garden* in SIF resolution (progressively scanned, 352×240, 30fps) and *Mobile*, *Foreman*, and *Coastguard* in CIF resolution (progressively scanned, 352×288, 30fps) were used for these results.

A. Motion Field

In Fig. 6, we show motion fields obtained by bidirectional and unidirectional motion estimation. We can see that in the areas (immediate right of the tree), the bidirectional motion field does not have such a messy distribution as the unidirectional motion field does, because for these areas in the corresponding frame we can find matched blocks with the same physical meaning in its following frame instead of finding pixels with similar luminance components at miss-matched places in its preceding frame. The influence of this more accurate motion field is threefold: decreased energy in temporal high frequency frames; classification of the connected and unconnected pixels is more consistent with the real occurrence of the occlusion effect, leading to temporal low frequency frames with better visual quality; and, lastly, fewer bits to encode the resulting smoother motion field. In this manner, we can have both efficient video coding and very good visual quality for the low frame-rate videos, almost avoiding all motion artifacts in the 1/2 and 1/4 frame-rate sequences.

B. Temporal Low Frequency Frames

In conventional MCTFs, some poorly matched pixels in frame two take part in the temporal filtering, causing motion artifacts in the low frame-rate video. Fig. 7 shows two temporal low subband frames, which are four temporal levels down. In Fig. 7(a) which is generated by unidirectional MCTF without using unconnected blocks, most of the motion artifacts come from the filtering across mismatched pixels. Fig. 7(b) which is generated by our new bidirectional MCTF incorporating unconnected blocks shows much fewer such artifacts and is more visually appealing as a low frame-rate version of the original video.

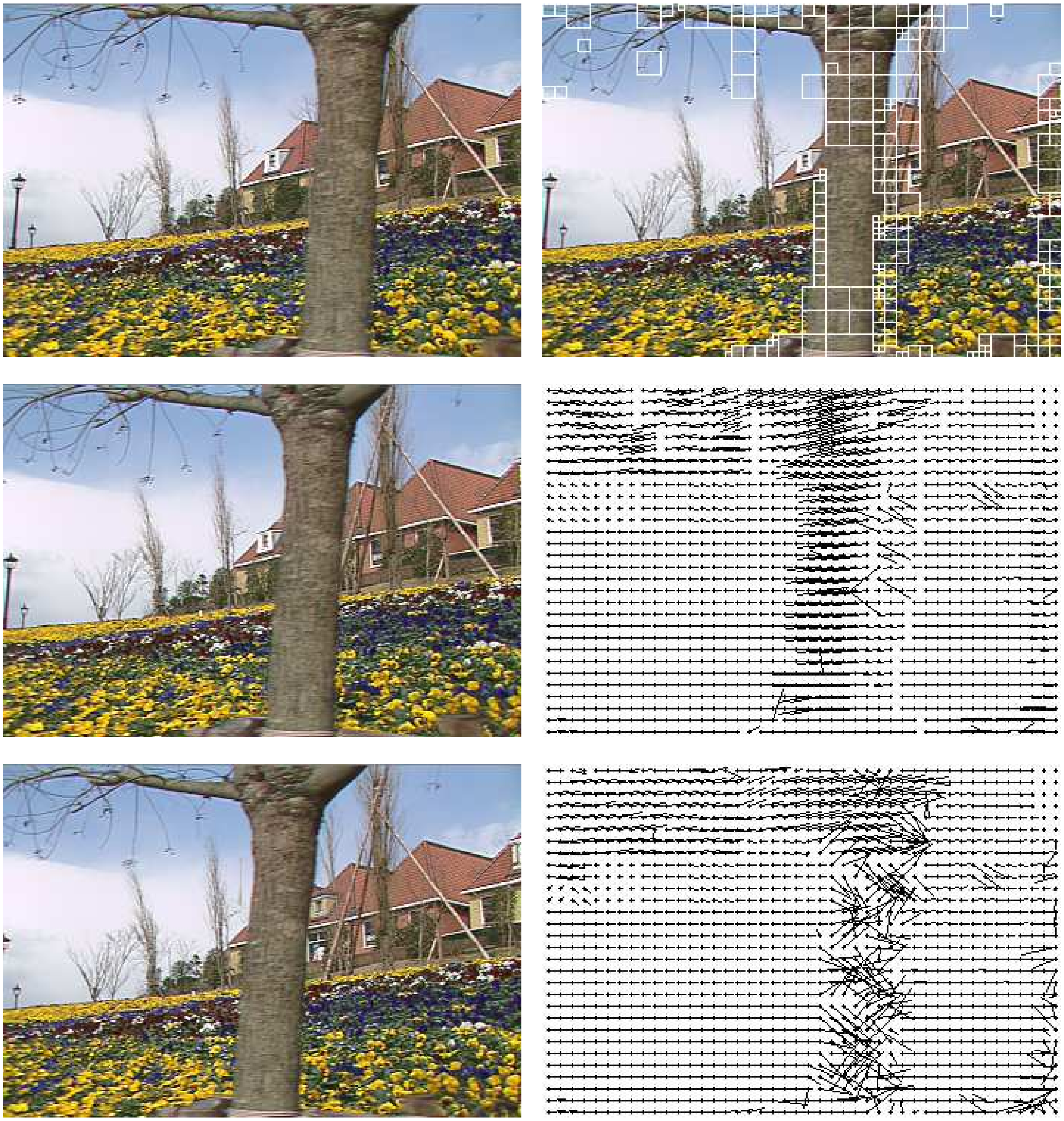
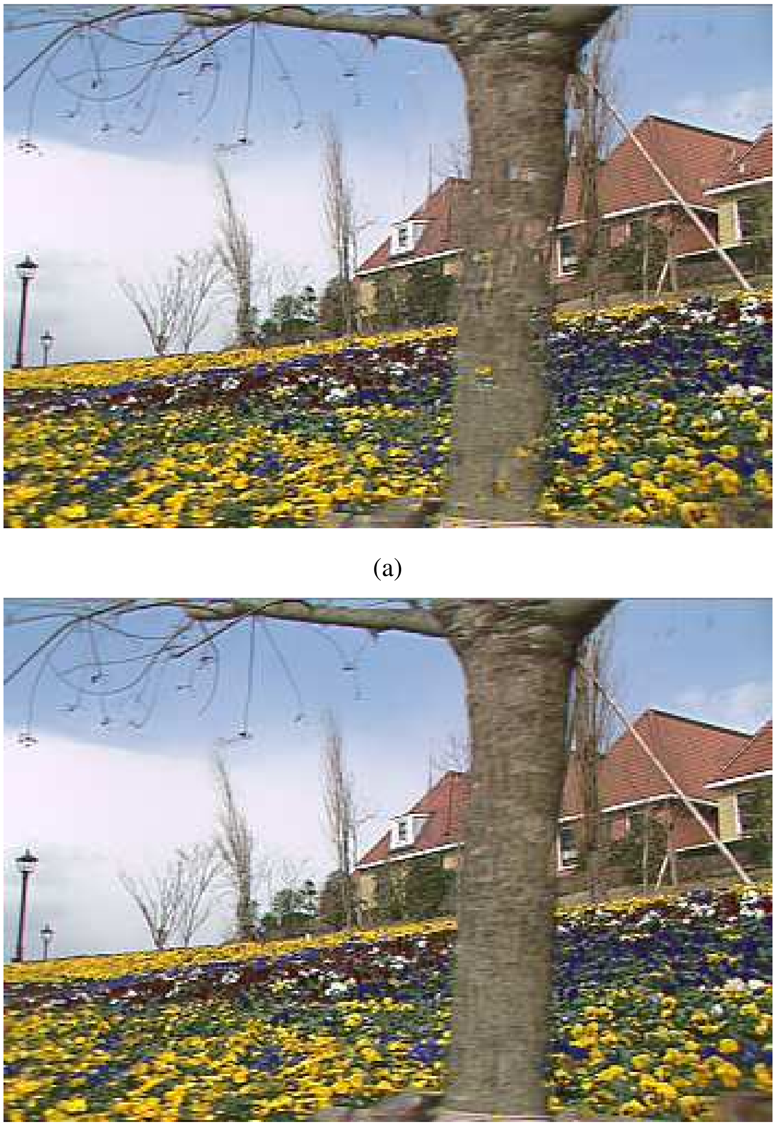


Fig. 6. Motion fields for *Flower Garden*. In the left column, from the top to the bottom are three consecutive frames two temporal levels down. In the right column, the first figure shows the unconnected blocks of the frame in the middle, the second is the motion field of the frame in the middle obtained by bidirectional motion estimation, and the third is the motion field obtained by unidirectional motion estimation.



(a)

(b)

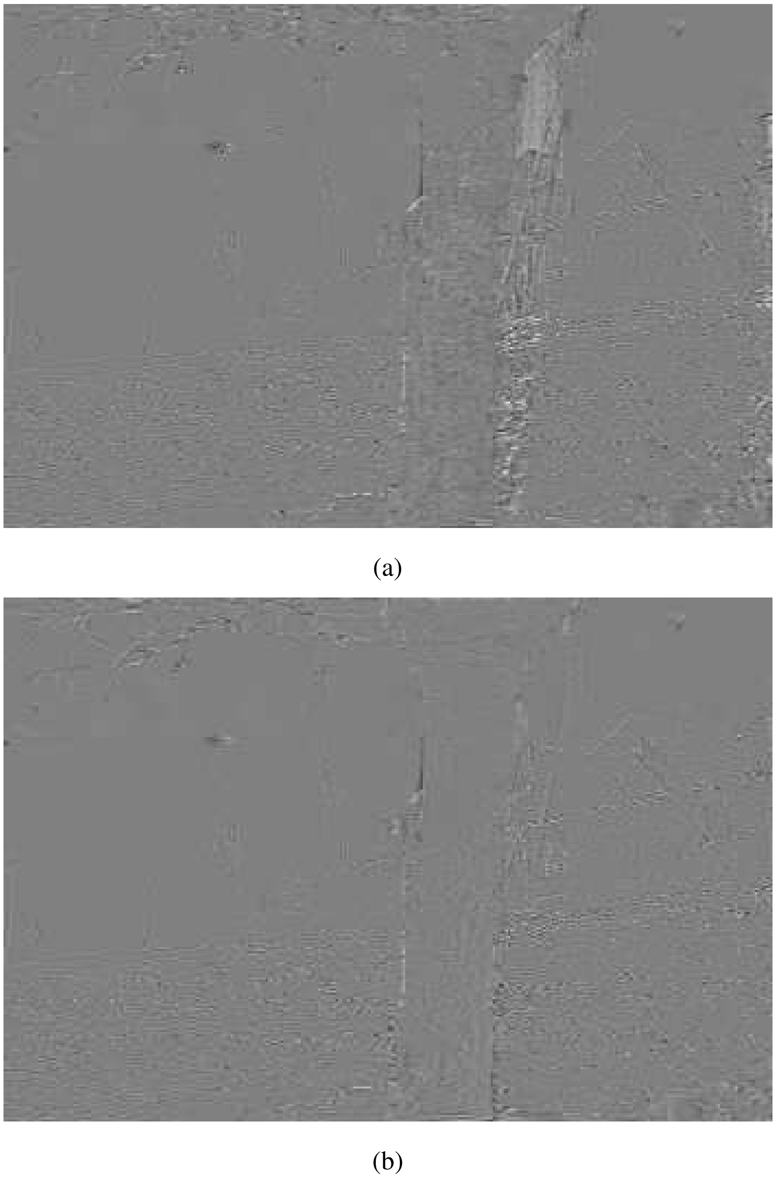
Fig. 7. Temporal low frequency frame t-LLLL1: (a) generated by unidirectional MCTF, (b) generated by bidirectional MCTF incorporating unconnected blocks.

C. *Temporal High Subbands*

Since we predict the unconnected blocks in frame two with a better matched reference, the energy in the temporal high subbands should decrease. This difference can be seen from Fig. 8 clearly, where $\frac{1}{4}$-pixel accurate MCTF is used. So in bidirectional MCTF, the energy is more concentrated in the temporal lower frequency subbands.

In Table II, we show the variance of temporal subbands generated by MCTF with different MV precisions. We can see with increasing MV precision, the energy in the temporal high subbands decreases dramatically.

We can also see this clearly from Fig. 9, where the values are scaled up and shifted positive for presentation.



(a)

(b)

Fig. 8. Temporal high frequency frame LLH1. (a) uni-directional MCTF; (b) bidirectional MCTF.

D. Coding Results

We next take a look at the influence of sub-pixel interpolation on compression. We compare the bilinear filter and our designed 8-tap FIR filters for $\frac{1}{2}$-pixel accurate MC-EZBC. From the coding results on *Mobile* as shown in Fig. 10, we can see that 8-tap FIR filters are much better than the bilinear filter. Fig. 11 shows the rate-distortion curves of MC-EZBC with different motion accuracies, with the reconstructed frames shown in Fig. 12. We can see quarter-pixel accurate MCTF has the fewest ringing artifacts. We also found upon viewing the videos that the ringing artifacts of integer-pixel and $\frac{1}{2}$-pixel accurate MCTF are more annoying than $\frac{1}{4}$-pixel accurate MCTF.

With the increase of MV accuracy, the computational complexity also increases, especially in the motion estima-

temporal subband variances	integer-pixel	$\frac{1}{2}$-pixel	$\frac{1}{4}$-pixel
H	101	48	36
LH	205	98	71
LLH	407	209	154
LLLH	797	442	341
LLLL	60009	61728	62048

TABLE II

TEMPORAL SUBBAND VARIANCES FOR THE LUMINANCE COMPONENT OF *Mobile* AT DIFFERENT MV PRECISIONS.

	Additions	Multiplications
Integer	$161N$	$8N$
Half	$265N$	$32N$
Quarter	$441N$	$128N$
1/8	$855N$	$512N$

TABLE III

COMPUTATIONAL COMPLEXITY OF MOTION ESTIMATION FOR ONE MOTION FIELD AT DIFFERENT MV ACCURACY, WHERE N IS THE TOTAL NUMBER OF PIXELS OF A FRAME.

tion stage. In our program, we have 5-level HVSBM and the sizes of our square motion blocks range from 4×4 to 64×64. Table III indicates the computational complexity of motion estimation for one field at different MV accuracies.

We also compare our MC-EZBC with H.26L (TML 9.0) [6]. We assume a broadcast or storage application, where both encoders (to make them comparable in efficiency) would use a GOP size of 16 (IBBBPBBBP. . . for H.26L, and four levels of temporal subband decomposition for MC-EZBC). Both coders use $\frac{1}{8}$-pixel accurate motion estimation and the same search region. The search region for *Mobile*, *Flower Garden*, and *Coastguard* is 16, but for *Foreman* is 32. In the H.26L software, a slight modification was necessary in order to prevent multiframe prediction across a GOP border, but was still possible within the GOP [16]. The H.26L codec was run with “all features on” at different QP levels, each encoding separately and internally optimized for the desired bit rate.

The MC-EZBC coder was run just once per sequence, and all the results at different bit rates were achieved by extracting various amounts from this one bitstream and then decoding. Rate-distortion curves are shown in Fig. 13-16. For *Mobile*, MC-EZBC achieves better objective results than H.26L (TML 9.0), and for *Flower Garden* and *Coastguard*, MC-EZBC’s objective performance is only slightly better, but for *Foreman*, MC-EZBC is worse than

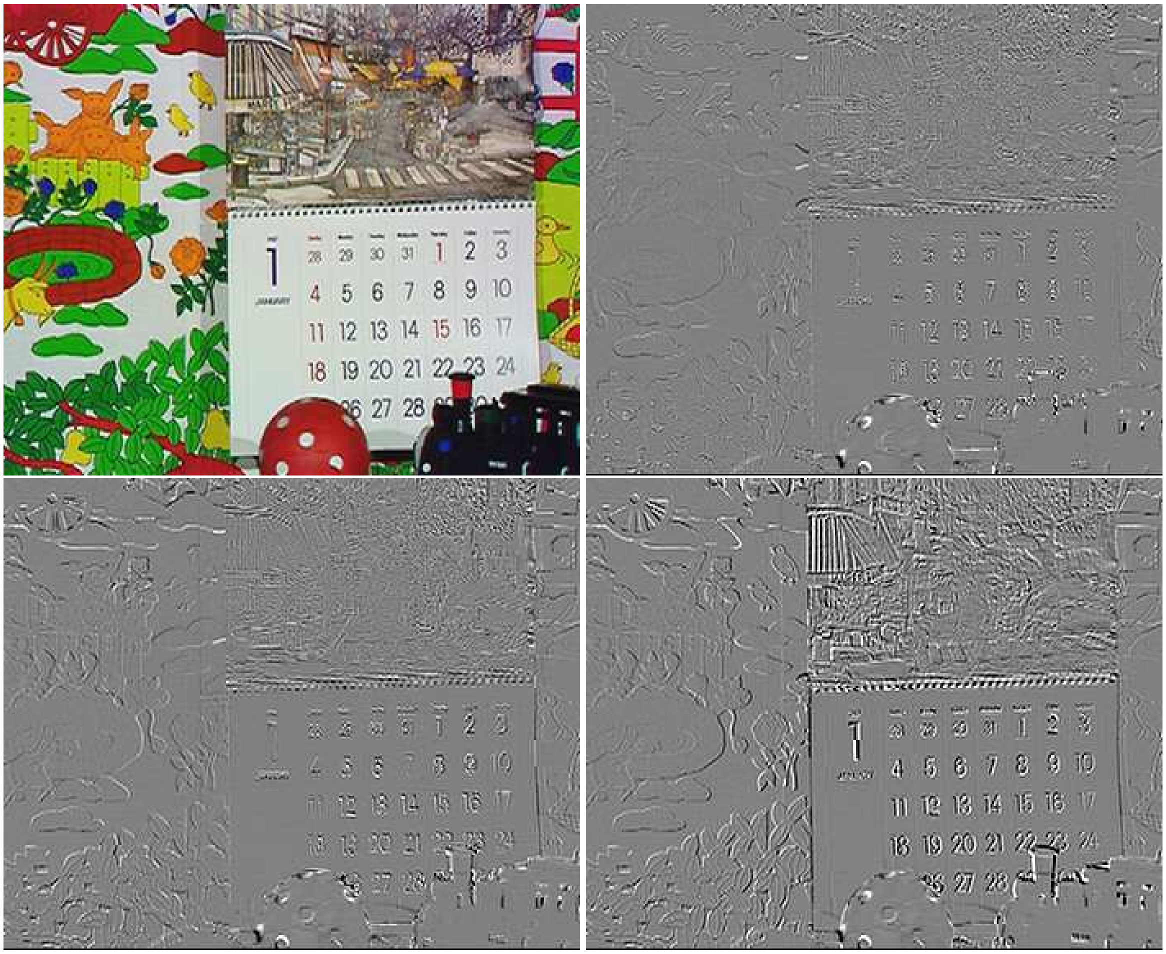


Fig. 9. First level temporal high subband frames generated from one level MCTF with different MV precisions. (a) top left: original frame, (b) top right: $\frac{1}{4}$-pixel MCTF, (c) bottom left: $\frac{1}{2}$-pixel MCTF, (d) bottom right: integer-pixel MCTF.

H.26L up to 2 dB. We think this last result is because *Foreman* contains a lot of nonrigid motion which cannot be handled well by our block-based motion model.

E. Quality and Temporal Scalability

Since MC-EZBC makes use of bit-plane coding, quality or PSNR scalability can be realized easily. In the experiments introduced in the last subsection, the MC-EZBC encoder was run just once per sequence, and all the results at different rates were achieved by accordingly decoding from this one bitstream, while H.26L encoding can only generate one bit stream for one result each time. Since H.26L (TML 9.0) optimizes each encoder pass, by comparing the rate-distortion curves generated by these two coding algorithms, we can find whether the scalability in MC-EZBC causes any performance loss or not. From the curves shown in Fig. 13 -16, we find the scalability behavior of MC-EZBC appears to be quite favorable, achieving approximately the rate-distortion performance of H.26L.

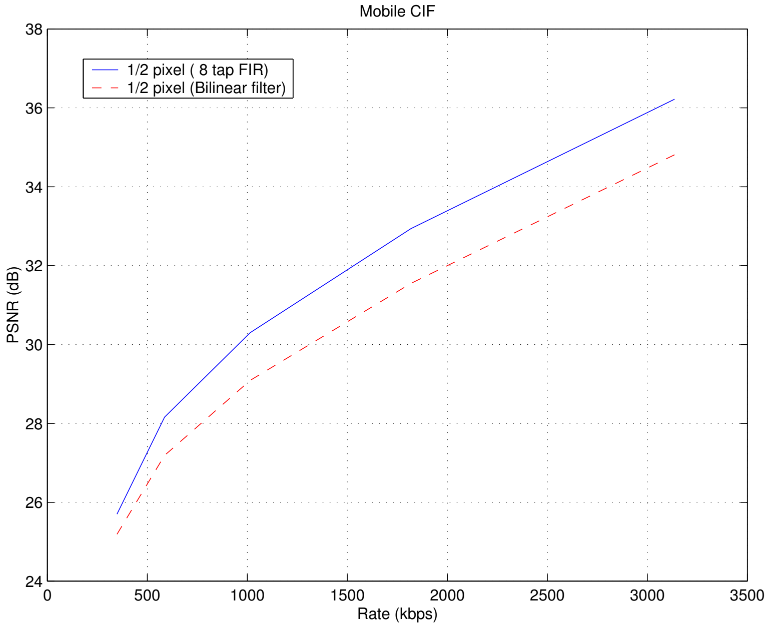


Fig. 10. Comparison of different interpolation filters.

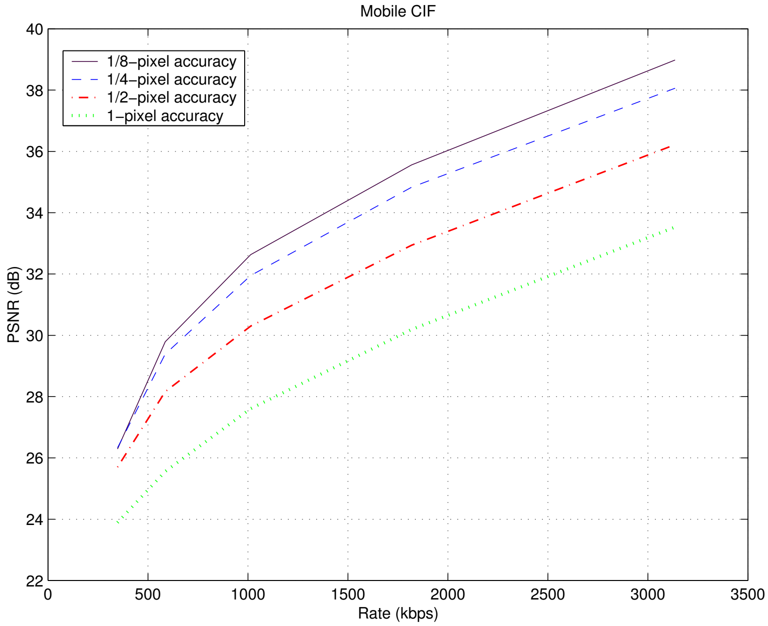


Fig. 11. Comparison of 1-pixel, $\frac{1}{2}$-pixel, $\frac{1}{4}$-pixel and $\frac{1}{8}$-pixel accuracy.

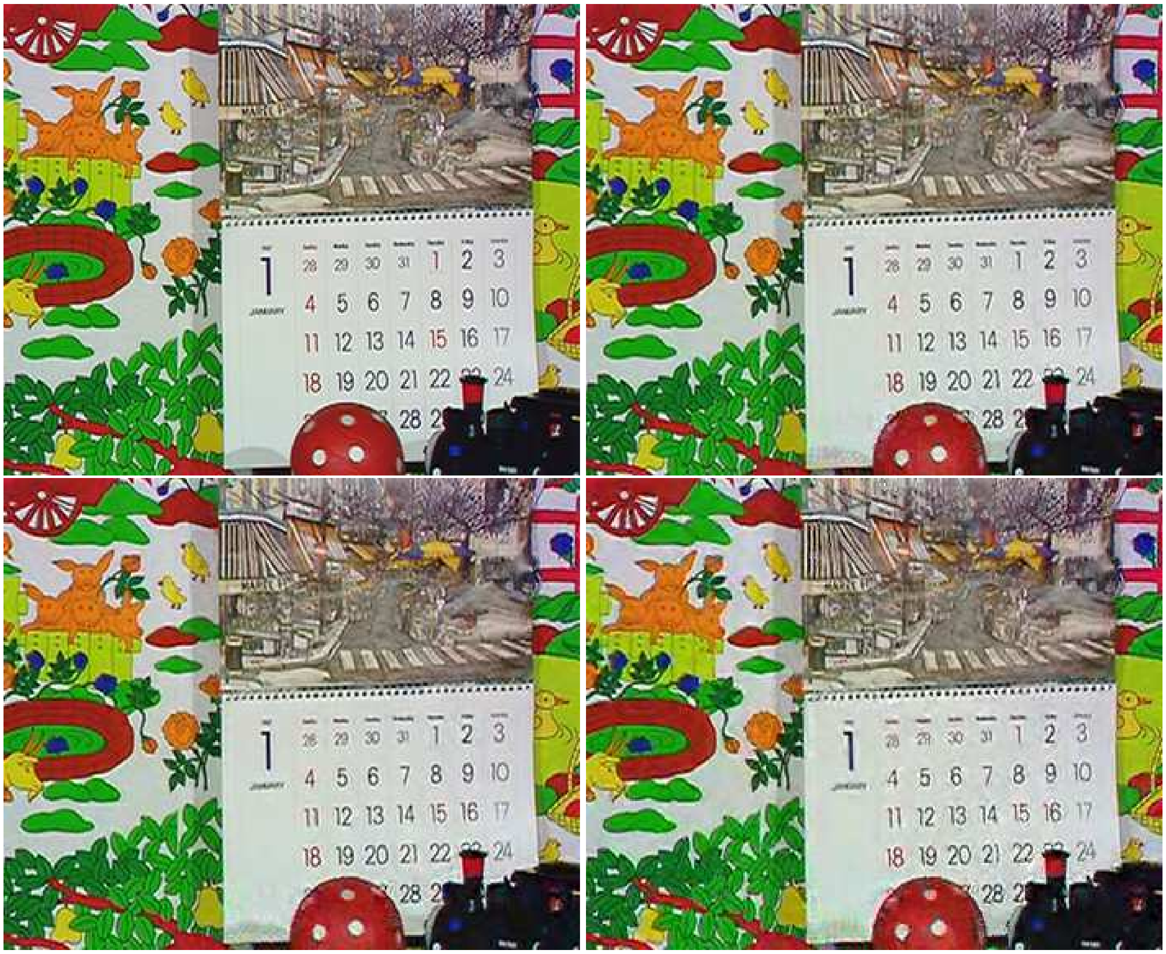


Fig. 12. Comparison of reconstructed frames coded at 586 kbps with different accurate MCTFs: (a) top left: original frame, (b) top right: $\frac{1}{4}$-pixel accuracy, Y-PSNR = 28.4 dB, (c) bottom left: $\frac{1}{2}$-pixel accuracy, Y-PSNR = 27.3 dB, (d)bottom right: integer-pixel accuracy, Y-PSNR = 25.2 dB.

Another advantage of MC-EZBC with bidirectional MCTF is for temporal scalability. By decoding different numbers of temporal subbands, several different frame rates can be offered at the receiver. Unidirectional MCTF has been seen to create some motion artifacts in the lower frame-rate sequences. The lower the frame rate, the more artifacts we get. Furthermore, since part of the information of the poorly matched blocks located in the second frame of a frame pair is saved in the temporal high subband frame in the unidirectional case, motion-compensated up-sampling to full frame rate, i.e. just use the low frame-rate video together with the MVs at higher levels, may not work very well. In bidirectional MCTF, most of the information of the unconnected blocks is contained in the temporal low subbands, which results in better MC up-sampling performance. We illustrate such MC up-sampling on *Flower Garden*. We first drop the temporal high-frequency subbands in the first and second temporal levels from a 1.8 Mbps bitstream. This saves about 42% of the bandwidth in transmission for both bidirectional and unidirectional MC-EZBC. Then, we reconstruct the full frame-rate sequence by MC up-sampling. We can see visual improvement with this approach by looking at the regions near the moving tree in Fig. 17.

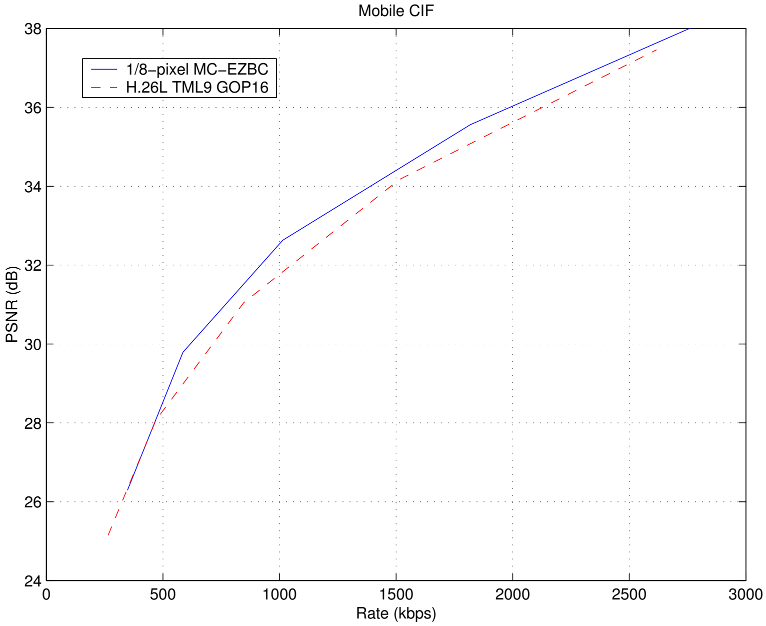


Fig. 13. Comparison of MC-EZBC and H.26L (TML 9.0) for CIF *Mobile*.

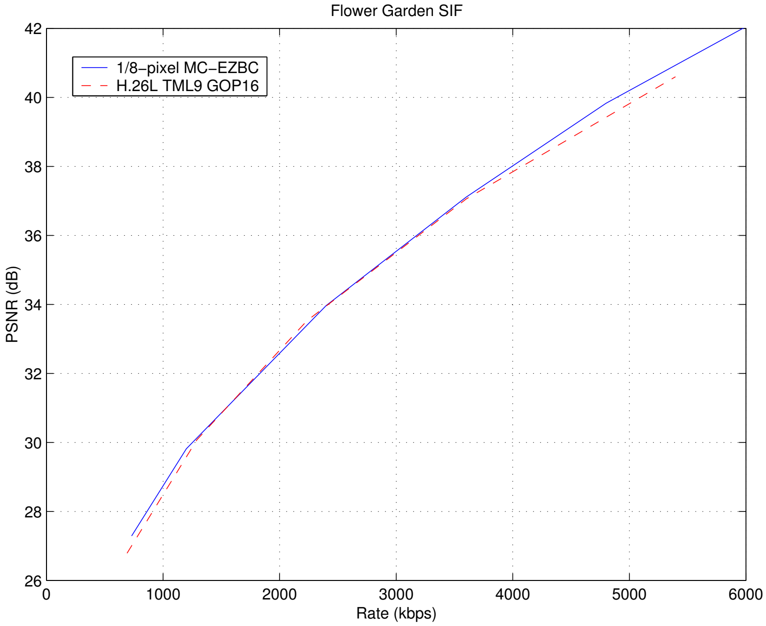


Fig. 14. Comparison of MC-EZBC and H.26L (TML 9.0) for SIF *Flower Garden*.

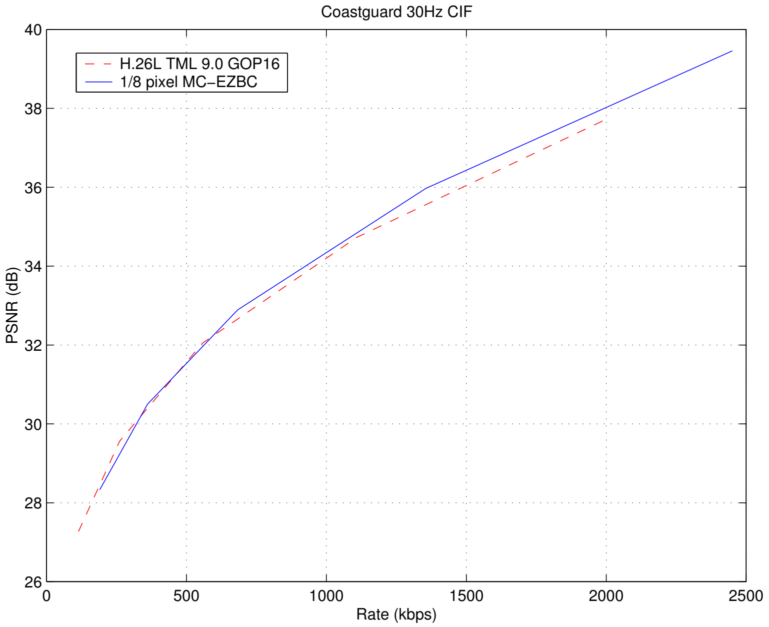


Fig. 15. Comparison of MC-EZBC and H.26L (TML 9.0) for CIF *Coastguard*.

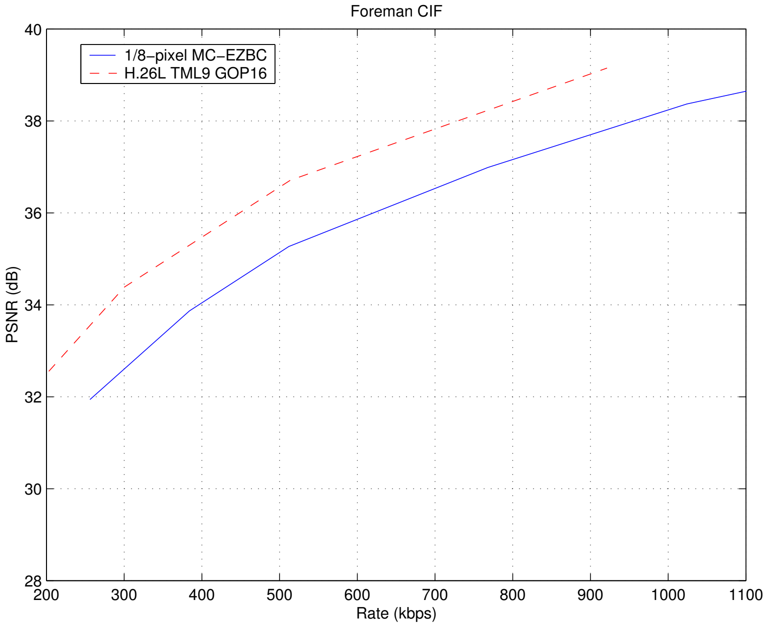
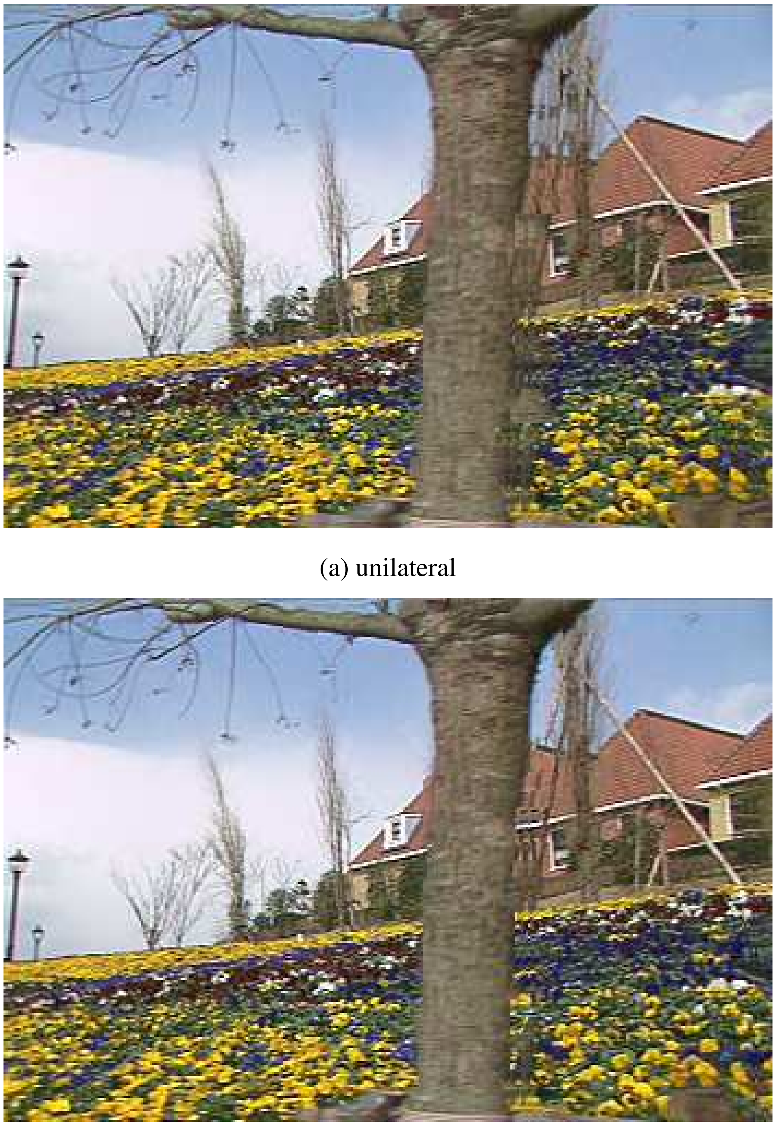


Fig. 16. Comparison of MC-EZBC and H.26L (TML 9.0) for CIF *Foreman*.



(a) unilateral

(b) bilateral

Fig. 17. MC up-sampled frame 4 of *Flower Garden* without transmitting the first and second level temporal high frequency subbands of the full rate 1.8 Mbps bitstream: (a) unilateral, (b) bilateral.

VIII. Conclusions

In this paper a new bidirectional MC-EZBC coder was constructed using a recently suggested lifting filter implementation, to introduce arbitrary sub-pixel accuracy into MCTF, while retaining invertibility for arbitrary interpolation methods in the absence of quantization. We addressed the question of best interpolation to use for this purpose. We also more carefully considered the quality of match in MCTF, and showed how it influenced both coding efficiency and the quality of the embedded low frame-rate video. We then designed algorithms to detect unconnected blocks, and introduced *I* blocks and *P* blocks for this purpose. Objective performance is on a par with

H.26L (TML 9) for several test videos, while offering the benefits of a fully embedded bit stream at seemingly no penalty.

REFERENCES

[1] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," *Applied and Computational Harmonic Analysis*, vol. 5, pp. 332-369, July 1998.

[2] P. Chen and J. W. Woods, *Comparison of MC-EZBC and H.26L TML 8 on Digital Cinema Test Sequences*, ISO/IEC JTC1/SC29/WG11, MPEG2002/8130, Cheju Island, March 2002.

[3] S. Choi and J. W. Woods, "Motion-compensated 3-D subband coding of video," *IEEE Trans. Image Processing*, vol. 8, no. 2, pp. 155-167, Feb. 1999.

[4] S. Choi, *Three-dimensional Subband/Wavelet Coding of Video with Motion Compensation*, PhD thesis, Rensselaer Polytechnic Institute, 1996.

[5] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, vol. 1, pp. 205-220, Apr. 1992.

[6] ITU-T, Video Coding Expert Group (VCEG), *H.26L test model long term number 9 (TML-9) draft 0*, 12/21/2001.

[7] Shih-Ta Hsiang and J. W. Woods, "Invertible three-dimensional analysis/synthesis system for video coding with half-pixel-accurate motion compensation," *Proc. VCIP'99*, SPIE, vol. 3653, Jan. 1999.

[8] Shih-Ta Hsiang and J. W. Woods, "Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling," *MPEG-4 Workshop and Exhibition at ISCAS 2000*, Geneva, Switzerland, May, 2000.

[9] Shih-Ta Hsiang and J. W. Woods, "Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank," *Signal Processing: Image Communication*, vol. 16, pp. 705-724, May, 2001.

[10] B. Haskell, A. Netravali, and A. Puri, *Digital Video: An Introduction to MPEG-2*, Kluwer Academic Pub., 1996.

[11] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, 1984.

[12] Ravi Krishnamurthy, John W. Woods and Pierre Moulin, "Frame interpolation and bidirectional prediction of video using compactly encoded optical-flow fields and label fields," *IEEE trans. on Circuits and Systems For Video Technology*, vol. 9, no. 5, pp. 713 -726, Aug. 1999.

[13] J.-R. Ohm, "Temporal domain subband video coding with motion compensation," in *Proc. ICASSP-92*, vol. 3, pp. 229-232, Mar. 1992.

[14] J.-R. Ohm, "Advanced packet-video coding based on layered VQ and SBC techniques," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 3, pp. 208-221, June 1993.

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

[16] J.-R. Ohm, K. Hanke, "*Principles for evaluation of scalable wavelet coding technology*," ISO/IEC JTC1/SC29/WG11, MPEG2002/8207, Cheju Island, March 2002.

[17] B. Pesquet-Popescu and V. Bottreau, "Three-dimensional lifting schemes for motion compensated video compression," *Proc. ICASSP*, pp. 1793-1796, May 2001.

[18] Boaz Porat, *A Course in Digital Signal Processing*, John Wiley & Sons, Inc., 1997.

[19] A. Secker and D. Taubman, "Motion-compensated Highly Scalable Video Compression Using An Adaptive 3-D Wavelet Transform Based on Lifting," *Proc. ICIP*, October 2001.