Subjective and Objective Comparison of Advanced Motion Compensation Methods for Blocking Artifact Reduction in a 3-D Wavelet Coding System

Cho-Chun Cheng, Wen-Liang Hwang, *Senior Member, IEEE,* Zuowei Shen, and Tao Xia

***Abstract*— We compare, both objectively and subjectively, the performance of various advanced motion compensation methods, including overlapped block motion compensation (OBMC) and control grid interpolation (CGI), in a 3-D wavelet-based coding system. The motion vectors of the methods are obtained by using a sequence of 1-D dynamic programming algorithms that minimizes the cost function. Our experiment results indicate that an OBMC sequence usually has a higher PSNR than other methods, while a CGI sequence usually contains the fewest blocking artifacts. We provide a simple framework that combines the OBMC and CGI sequences. The proposed hybrid method removes more than 50% of the blocking artifacts of an OBMC sequence, while simultaneously maintaining a high PSNR performance.**

***Index Terms*— Wavelet transform, overlapped block motion compensation (OBMC), control grid interpolation (CGI)**

I. Introduction

Recent advances in motion-compensated temporal filtering (MCTF) based on 3-D discrete wavelet transforms have proven that it is possible to implement highly efficient video codecs with spatial, temporal, and SNR scalability [12]. In most 3-D video coding systems, the block matching algorithm (BMA) is the predominant method used to reduce temporal redundancy because of its relatively low computational complexity. However, it is well known that the discontinuous motion field generated by BMA leads to unpleasant blocking artifacts in low bit-rate applications. Previous research on traditional hybrid video coding systems has suggested that an advanced motion compensation technique, characterized by inter-block dependency motion vectors like control grid interpolation (CGI) and overlapped block motion compensation (OBMC), can remove blocking artifacts and improve the visual quality of videos [14], [16]. A number of interesting systems that combine different motion compensation methods have also been proposed; for example, a hybrid of OBMC and BMA [9], CGI and BMA [5], and an image wrapping method and OBMC [4]. Techniques that reduce blocking artifacts in 3-D video coding systems have also been reported. The systems in [3] uses a filter to smooth the areas whose neighboring motion vectors are inconsistent. The early coding results of 3-D subband coding using CGI and OBMC are reported in [11] and [17], respectively. However, optimized motion vector estimation, which is the most crucial part of improving the quality of a motion compensation system, is not a trivial task and is not addressed in these papers.

We first present temporal filtering frameworks based on CGI and OBMC. Then, inspired by the work in [1], we propose a motion estimation algorithm that uses an iterative motion refinement procedure based on dynamic programming to solve the dependent motion estimation problem. The coding performance depends on the initial conditions and the iterative procedure for refining the motion vectors. Our focus is not to compare the performance of our implementation with that of other implementations [1], [4]. Instead, we concentrate on developing subjective and objective comparisons of various advanced motion compensation algorithms which aim to reduce the blocking artifacts in a 3-D video coding system. The subjective and objective coding results of our OBMC and CGI show that each method generates distinct side effects during the de-blocking process. Specifically, CGI usually contains the fewest blocking artifacts, but has a lower PSNR due to the stronger smoothing effect of a completely smoothed motion field. In contrast, OBMC usually achieves a higher PSNR, but poor de-blocking results often appear along block boundaries with inconsistent motion activities. Thus, we combine both systems (called OBCG hereafter) to obtain a sequence that yields low blocking artifacts and achieves high PSNR simultaneously. In OBCG, we use a simple decision mechanism to select one of the two modes of each block and use the PSNR cost function and dynamic programming to derive the optimal motion field. Each block of our OBCG method contains one MV, and it does not need to transmit side information, such as the mode decision of each block. Therefore, the decoder's complexity is the same as that of decoding an OBMC or a CGI sequence. A similar hybrid system that combines image wrapping and OBMC, proposed in [4], uses the quadtree to partition and determine the mode (image warping or OBMC) of each partition based on MSE minimization. It differs from our method in two respects: 1) our goal is to reduce blocking artifacts, whereas that of [4] is to minimize MSE; and 2) the system in [4] needs to send the quadtree partitions and modes to decoders as side information.

Manuscript received November 16, 2005; revised March 21, 2006. This work was supported in part by National Science Council, R.O.C., under contract NSC 95-2213-E-001-005. This paper was recommended by Associate Editor H. Gharavi.

C.-C. Cheng and W.-L. Hwang are with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan, R.O.C. (e-mail: ed-diecc,whwang@iis.sinica.edu.tw).

Z. Shen is with the Department of Mathematics, National University of Singapore, 117543, Singapore (e-mail: matzuows@nus.edu.sg).

T. Xia was with the Center for Wavelet, Approximation and Information Processing, National University of Singapore, 117543, Singapore (e-mail: tao.xia04@gmail.com).

Digital Object Identifier XXX.

The remainder of the paper is organized as follows. In Section II, we describe the temporal filtering frameworks of CGI and OBMC, the optimized motion estimation method, and report the coding results. In Section III, we propose our OBCG system, and report the subjective and objective performance results of the system. Then, in Section IV, we present our conclusions.

II. The Performance of OBMC and CGI in MCTF

The temporal filtering frameworks of OBMC and CGI are based on a series of works by Ohm and Woods et. al [10], [2]. First, let $f_t(\mathbf{s})$ denote the pixel intensity at spatial position $\mathbf{s} = (x, y)$ of frames f_t, where t is the temporal index. Under the MCTF framework, which is based on the $5-3$ filter, motion compensation is implemented by predicting odd frames from even ones. During motion compensation, f_{2t+1} is first partitioned into a group of non-overlapping square blocks, $\mathbf{P} = (P_{0,0}, \cdots, P_{N_1-1,N_2-1})$, each of which then finds its corresponding block with the least distortion measurement in f_{2t} and f_{2t+2} during the motion estimation operation. Let MC denote a BMA, CGI, or OBMC motion compensation technique. Also, let $\hat{f}^{MC-}_{2t+1}(\mathbf{s})$ and $\hat{f}^{MC+}_{2t+1}(\mathbf{s})$ respectively denote the backward and forward predicted pixel intensities at $\mathbf{s}$ in f_{2t+1} derived by the MC technique. The backward and forward motion vectors (MVs) are denoted as $\mathbf{v}^{MC-}(\mathbf{s})$ and $\mathbf{v}^{MC+}(\mathbf{s})$, and their inverse counterparts are, respectively, $\tilde{\mathbf{v}}^{MC-}(\mathbf{s}) = -\mathbf{v}^{MC-}(\mathbf{s})$ and $\tilde{\mathbf{v}}^{MC+}(\mathbf{s}) = -\mathbf{v}^{MC+}(\mathbf{s})$. The temporal highpass filtering scheme is given by

$$H_t(\mathbf{s}) = f_{2t+1}(\mathbf{s}) - \frac{1}{2}(\hat{f}^{MC-}_{2t+1}(\mathbf{s}) + \hat{f}^{MC+}_{2t+1}(\mathbf{s})), \quad \mathbf{s} \in P_{i,j}; \quad (1)$$

and the temporal lowpass filtering scheme for connected and multiple-connected pixels is

$$\begin{aligned} L_t(\mathbf{s}) = f_{2t}(\mathbf{s}) \quad &+ \quad \frac{1}{4}(H_{t-1}(\mathbf{s} + \tilde{\mathbf{v}}^{MC-}(\mathbf{s})) \\ &+ \quad H_t(\mathbf{s} + \tilde{\mathbf{v}}^{MC+}(\mathbf{s}))), \mathbf{s} \in P_{i,j}. \quad (2) \end{aligned}$$

Even though we require two sets of motion vector fields to predict f_{2t+1}, they can be derived by the same prediction technique. Thus, for simplicity, the following discussion only describes backward prediction, as forward prediction can be similarly derived.

Both CGI and OBMC try to eliminate unexpected discontinuities in motion-compensated frames. CGI produces a smooth motion field in the predicted frame by choosing four corner vertices of each block as control points, with each point assigned to an MV in the set $\mathbf{V}^{MC-} = \mathbf{U}^{-} \equiv (\mathbf{u}^{-}_{0,0}, \cdots, \mathbf{u}^{-}_{N_1,N_2})$. Let $\mathbf{u}^{-}_{i+p,j+q}(p, q = 0, 1)$ be the MVs associated with four control points in $P_{i,j}$. For each interior pixel in $P_{i,j}$, CGI computes an individual MV by interpolating the MVs of the control points. Let $w_{p,q}(\mathbf{s})$ denote the interpolation function of the (p, q)th control point, which indicates the desired contribution of the MV of the (p, q)th control point to $\mathbf{s}$, and $\sum_{p=0}^{1}\sum_{q=0}^{1} w_{p,q}(\mathbf{s}) = 1$. The filtering scheme of CGI can be formulated by replacing $\hat{f}^{MC-}_{2t+1}(\mathbf{s})$ and $\mathbf{v}^{MC-}(\mathbf{s})$ in (1) and (2) with $\hat{f}^{CGI-}_{2t+1}(\mathbf{s})$ and $\mathbf{v}^{CGI-}(\mathbf{s})$ respectively, given as

$$\hat{f}^{CGI-}_{2t+1}(\mathbf{s}) = f_{2t}(\mathbf{s} - \mathbf{v}^{CGI-}(\mathbf{s})), \quad (3)$$

$$\mathbf{v}^{CGI-}(\mathbf{s}) = \sum_{p=0}^{1}\sum_{q=0}^{1} w_{p,q}(\mathbf{s})\mathbf{u}^{-}_{i+p,j+q}. \quad (4)$$

In contrast to CGI, the OBMC scheme assigns one MV to each block. The intensity of each pixel is the weighted sum of the pixels predicted by using the MVs of its neighboring blocks. We denote the MV set as $\mathbf{V}^{MC-} = \mathbf{V}^{-} \equiv (\mathbf{v}^{-}_{0,0}, \cdots, \mathbf{v}^{-}_{N_1-1,N_2-1})$. Let $\alpha_{p,q}(\mathbf{s})$ be the weighting function at $\mathbf{s}$ using the MV of its (p, q)th neighboring block, and $\sum_{p=-1}^{1}\sum_{q=-1}^{1} \alpha_{p,q}(\mathbf{s}) = 1$. The filtering scheme of OBMC can then be formulated by replacing $\hat{f}^{MC-}_{2t+1}(\mathbf{s})$ and $\mathbf{v}^{MC-}(\mathbf{s})$ in (1) and (2) with $\hat{f}^{OBMC-}_{2t+1}(\mathbf{s})$ and $\mathbf{v}^{OBMC-}(\mathbf{s})$ respectively, given as

$$\hat{f}^{OBMC-}_{2t+1}(\mathbf{s}) = \sum_{p=-1}^{1}\sum_{q=-1}^{1} \alpha_{p,q}(\mathbf{s}) f_{2t}(\mathbf{s} - \mathbf{v}^{-}_{i+p,j+q}), \quad (5)$$

$$\mathbf{v}^{OBMC-}(\mathbf{s}) = \mathbf{v}^{-}_{i,j}, \quad \forall \mathbf{s} \in P_{i,j}. \quad (6)$$

A. Motion Vector Estimation and Complexity

The purpose of estimating optimal motion vectors is to find those that minimize a cost function. Let $C^{MC}(P_{i,j}, V^{MC}(P_{i,j}))$ denote the cost of estimating the coefficients in $P_{i,j}$ by using a motion compensation scheme MC. The cost of $P_{i,j}$ is determined by the MVs of a set of control points, represented by $V^{MC}(P_{i,j})$. Let $\mathbf{X}$ be the motion vectors of the image. The motion vector estimation problem can be formulated as finding $\mathbf{X}^{*}$ that minimizes the following cost function

$$\begin{aligned} \mathbf{X}^{*} &= arg \min_{\mathbf{X}} J \\ &= arg \min_{\mathbf{X}} \{ \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} C^{MC}(P_{i,j}, V^{MC}(P_{i,j})) \}, (7) \end{aligned}$$

where

$$\begin{aligned} & C^{MC}(P_{i,j}, V^{MC}(P_{i,j})) \\ = \quad & \frac{1}{\sharp P_{i,j}} \{ \sum_{\mathbf{s} \in P_{i,j}} |f(\mathbf{s}) - \hat{f}^{MC}(\mathbf{s})| + \lambda R(V^{MC}(P_{i,j})) \}, (8) \end{aligned}$$

where $\sharp P_{i,j}$ represents the number of pixels in $P_{i,j}$. If MC is CGI, then $\mathbf{X}$ is $\mathbf{U}$ and $V^{CGI}(P_{i,j}) = \{\mathbf{u}_{i+p,j+q} \mid p = 0, 1; q = 0, 1\}$; however, if MC is OBMC, then $\mathbf{X}$ is $\mathbf{V}$ and $V^{OBMC}(P_{i,j}) = \{\mathbf{v}_{i+p,j+q} \mid p = -1, 0, 1; q = -1, 0, 1\}$.

Estimating the optimal dependent motion vectors of an image is a 2-D problem that requires an exponential number of computational complexity [1]. Therefore, a tradeoff between performance and complexity is necessary. We propose an iterated algorithm that alternately modifies the motion vectors of even rows, while those of the odd rows are fixed, and vice versa. Let $\mathbf{X} = \mathbf{X}_e \bigcup \mathbf{X}_o$, where $\mathbf{X}_e = \{\mathbf{X}_{2i}\}$ and $\mathbf{X}_o = \{\mathbf{X}_{2i+1}\}$ denote the motion vectors on the even and odd rows respectively. We attempt to find a set of optimal MVs $\mathbf{X}^{*} = \mathbf{X}^{*}_e \bigcup \mathbf{X}^{*}_o$ by the following derivations

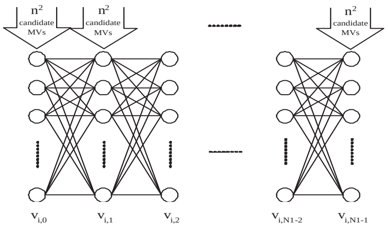


Fig. 1. An example of 1-D row processing of the ith row based on dynamic programming.

$$
\begin{aligned}
\min_{\mathbf{X}} J &= \min_{\mathbf{X}^e} J_e(\mathbf{X}_e|\, \mathbf{X}_o^*) + \min_{\mathbf{X}_o} J_o(\mathbf{X}_o|\, \mathbf{X}_e^*) \\
&\leq \min_{\mathbf{X}_e} J_e(\mathbf{X}_e|\, \mathbf{X}_o^{'}) + \min_{\mathbf{X}_o} J_o(\mathbf{X}_o|\, \mathbf{X}_e^{'}) \\
&= \sum_k \min_{\mathbf{X}_{2k}} J_{2i}(\mathbf{X}_{2k}|\, \mathbf{X}_o^{'}) \\
&+ \sum_k \min_{\mathbf{X}_{2k+1}} J_{2i+1}(\mathbf{X}_{2k+1}|\, \mathbf{X}_e^{'}), \qquad (9)
\end{aligned}
$$

where $\mathbf{X}_e^{'}, \mathbf{X}_o^{'}$ are arbitrary motion vectors, and $J_e(J_{2k})$ is defined as in (7); however, i in (7) is restricted to even numbered rows (at the $2k$-th row), and $J_o(J_{2k+1})$ restricts i to odd numbered rows (at the $2k+1$-th row). The last equality in (9) shows that the 2-D minimization problem in (7) is converted to a series of 1-D minimization problems. Meanwhile, the first inequality in the equation shows that when the given motion vectors of the even or odd rows are not the optimal vectors, the distortion is larger than the optimal solution. However, when $\mathbf{X}_o' = \mathbf{X}_o^*$ and $\mathbf{X}_e' = \mathbf{X}_e^*$, we can derive the optimal solution. This observation inspires an iterative algorithm that repeatedly updates the result of the previous iteration of even rows followed by odd rows. The proposed algorithm is:

```
**Step 1** Initialize the motion vectors of each row $\mathbf{X}_i^0 = \{v_{i,j}^0,\, j = 0, \cdots, N_m - 1\}$. Let $\mathbf{X}_e^0 = \bigcup_i \mathbf{X}_{2i}^0, \mathbf{X}_o^0 = \bigcup_i \mathbf{X}_{2i+1}^0, \mathbf{X}^0 = \mathbf{X}_e^0 \bigcup \mathbf{X}_o^0$, and let $k = 1$.
**Step 2** Obtain $\mathbf{X}_e^k$, and $\mathbf{X}_e^k = \arg\min_{\mathbf{X}_e} J_e(\mathbf{X}_e|\, \mathbf{X}_o^{k-1}) = \bigcup_i \arg\min_{\mathbf{X}_{2i}} J_{2i}(\mathbf{X}_{2i}|\, \mathbf{X}_o^{k-1})$.
**Step 3** Obtain $\mathbf{X}_o^k$, and $\mathbf{X}_o^k = \arg\min_{\mathbf{X}_o} J_o(\mathbf{X}_o|\, \mathbf{X}_e^k) = \bigcup_i \arg\min_{\mathbf{X}_{2i+1}} J_{2i+1}(\mathbf{X}_{2i+1}|\, \mathbf{X}_e^k)$.
**Step 4** Let $\mathbf{X}^k = \mathbf{X}_e^k \bigcup \mathbf{X}_o^k$, $k \leftarrow k + 1$, and go to **Step 2** until $|\mathbf{X}^k - \mathbf{X}^{k-1}| < \epsilon$.
```

Our algorithm modifies the MVs of each row, whereas in [1], the MVs of the rows and columns are modified alternately. We now analyze the computational complexity of various motion estimation techniques as follows. Let n be the square search window size. BMA requires $N_1 N_2 n^2$ computations to obtain its MV set. Figure 1 illustrates 1-D row processing by using dynamic programming in **Steps 2** and **3**, where each column of nodes corresponds to the MV candidates for a block; and each link between adjacent columns of nodes represents the distortion of the dependency. Therefore, n^4 computations for the links between adjacent columns are required for n^2 nodes in each column. Accordingly, our algorithm requires $l(N_1 - 1)N_2 n^4$ computations, where l is the number of iterations. Our experiments show that 2 to 4 iterations are sufficient to achieve convergence in most sequences. Compared to the $\min\{n^{4N_1}(N_2 - 1), n^{4N_2}(N_1 - 1)\}$ computations required to obtain the optimum solution of the 2-D problem, the complexity of our algorithm is significantly lower.

B. Objective and Subjective Evaluations

We now compare the objective and subjective performances of the BMA, OBMC, and CGI coding schemes. The objective performance is measured by the Y-PSNR, while the subjective evaluation is based on the DSIS (Double Stimulus Impairment Scale) measurement, which is designed according to the requirements of [6]. The participants were required to grade the sequences using a five-grade impairment scale from imperceptible (5) to very annoying (1). BMA was implemented using a full-search with an integer-pel accuracy. The block size was 16×16, and the search range for both the horizontal and vertical dimensions was $[-16, 15]$. The rate-distortion optimization parameter, $\lambda = 50$, was empirically determined. For CGI and OBMC, we use a full-search BMA to obtain the initial MVs because its complexity is less than that of the method in [1]. The motion field is then updated by our algorithm. During motion refinement, CGI uses the bilinear transform to obtain sub-pel accuracy motion vectors, which are then rounded to integer-pel accuracy. Our OBMC uses the 24×24 weighting window in [13]. The MV rate used in (8) is estimated by the universal variable length code (UVLC) proposed in H.264, and all motion vectors are differentially encoded as the standard suggests. First, three levels of temporal filtering of the $5 - 3$ filter are applied, followed by three levels of spatial wavelet transform with the $9 - 7$ filter. Finally, we use 3-D SPIHT (Set Partitioning in Hierarchical Trees) [7] to encode the wavelet coefficients.

In Figure 2, the curves labeled BM-CGI, are the performance results of CGI using BMA to obtain the MVs. Compared to the performance of BMA, the average PSNR degradation of BM-CGI is significant. As shown by the curves labeled DP-CGI, whose MVs are derived by our algorithm, the PSNRs are significantly better (0.5 - 1.3 dB) than those of BM-CGI. The results of OBMC in Figure 2, including BM-OBMC (MVs obtained by BMA) and DP-OBMC (MVs obtained by our algorithm), demonstrate a considerable PSNR performance improvement over BMA. The coding gain is 0.2 - 0.5 dB on average. Given the small PSNR improvement of DP-OBMC over BM-OBMC, we conclude that OBMC is relatively insensitive to the accuracy of the motion field. Figure 3 shows the subjective evaluation results of various motion compensation methods at low bit-rates; News was encoded at 100 kbits/sec, Silent at 125 kbits/sec, Salesman at 125 kbits/sec, Dancer at 200 kbits/sec, Foreman at 350 kbits/sec, and Mother and Daughter at 200 kbits/sec. The scores were obtained by averaging the individual scores of 15 participants. The results show that both DP-CGI and DP-OBMC achieve better perceptual quality of various video sequences compared to that obtained by BMA.

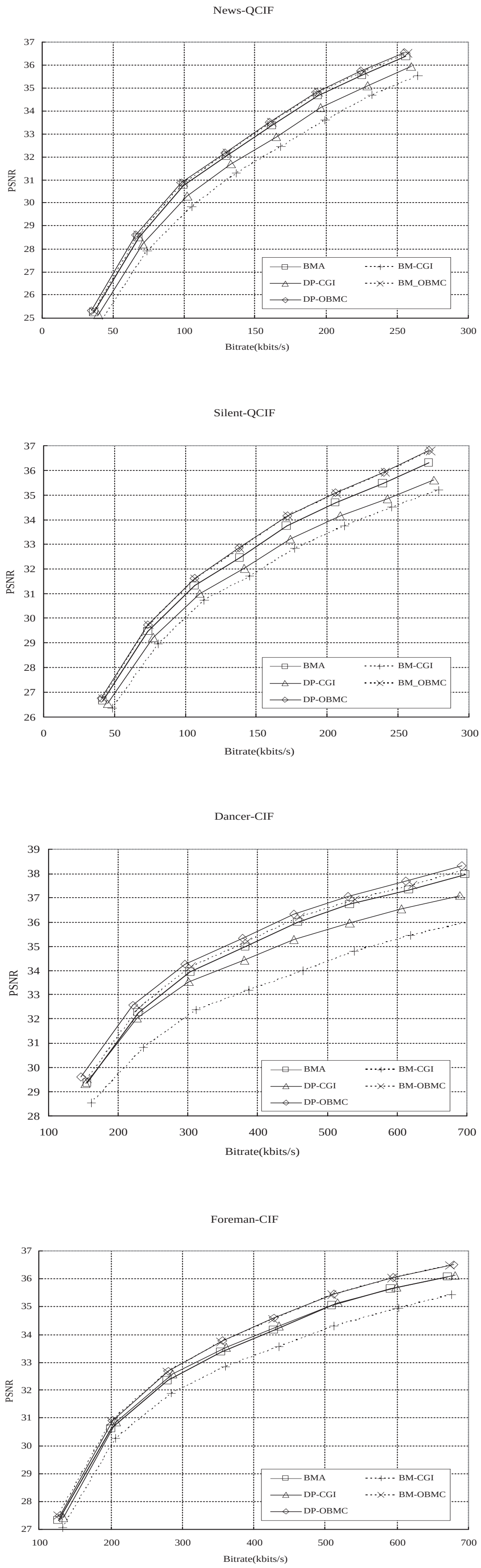


Fig. 2. The PSNRs of BMA, BM-CGI, DP-CGI, BM-OBMC, and DP-OBMC at various bit-rates (including MV rates) for different video sequences and formats.

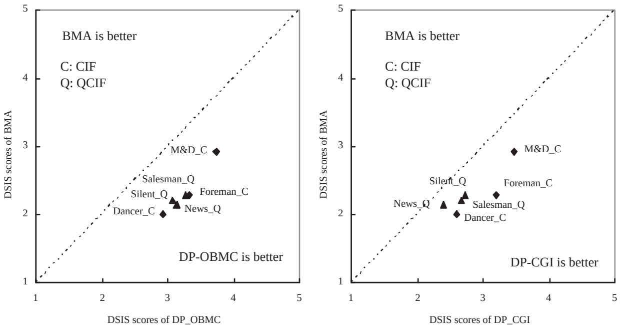
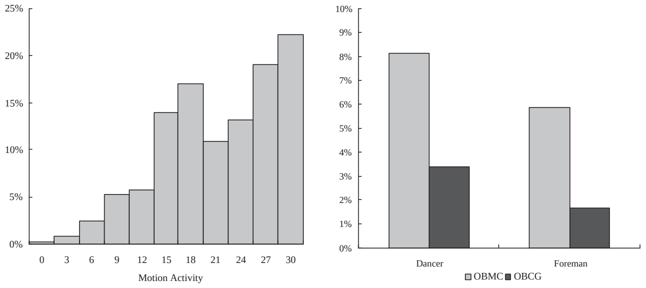


Fig. 3. Perceptual comparison of BMA, DP-CGI, and DP-OBMC.

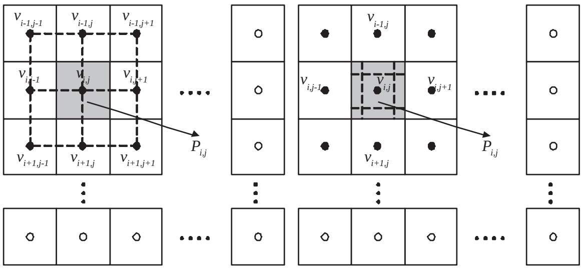


(a) (b)

Fig. 4. (a)The average percentage of blocks with blocking artifacts versus motion activity in the *Dancer* and *Foreman* OBMC sequences. (b) Comparison of the average percentage of blocks with blocking artifacts in the OBMC and OBCG sequences.

III. A Hybrid OBMC/CGI Scheme

Our objective results indicate that, in general, OBMC achieves a better PSNR than CGI. However, the artifact patterns of the methods are different. Although CGI sequences usually contain fewer blocking artifacts, they tend to over-smooth the edges or object boundaries. OBMC sequences, on the other hand, remove some, but not all, blocking artifacts. In many parts of the OBMC sequences, we can clearly observe blocking artifacts. To further substantiate our observation, we asked our testers to identify, frame-by-frame, the blocks that contained blocking artifacts after coding by OBMC. The plot in Figure 4 shows the average percentage of blocks with blocking artifacts versus the motion activity of various OBMC sequences. The motion activity was measured by calculating the maximal absolute difference between the motion vector of a block and the motion vectors of its neighboring blocks. We observe that blocking artifacts are still very likely to occur in areas where motion activity is high. Inspired by the approach in [8], which selects the mode at the decoder, we propose a mode selection approach that removes blocking artifacts during motion compensation at the encoder. Our proposed approach, which is straightforward, combines the advantages of the higher PSNR of OBMC and fewer blocking artifacts of CGI.



(a) (b)

Fig. 5. Each block is assigned one MV in OBCG. (a) shows that if the marked block is to be compensated for by CGI, then the MVs at the corners of the block are obtained from the neighboring MVs of the block. For example, the MV of the upper left-hand corner is obtained from $v_{i-1,j-1}$, $v_{i-1,j}$, $v_{i,j-1}$, and $v_{i,j}$ by bilinear interpolation. (b) shows the implementation of the case if the marked block is compensated for by OBMC.

A. Mode Decision and Optimized Motion Estimation

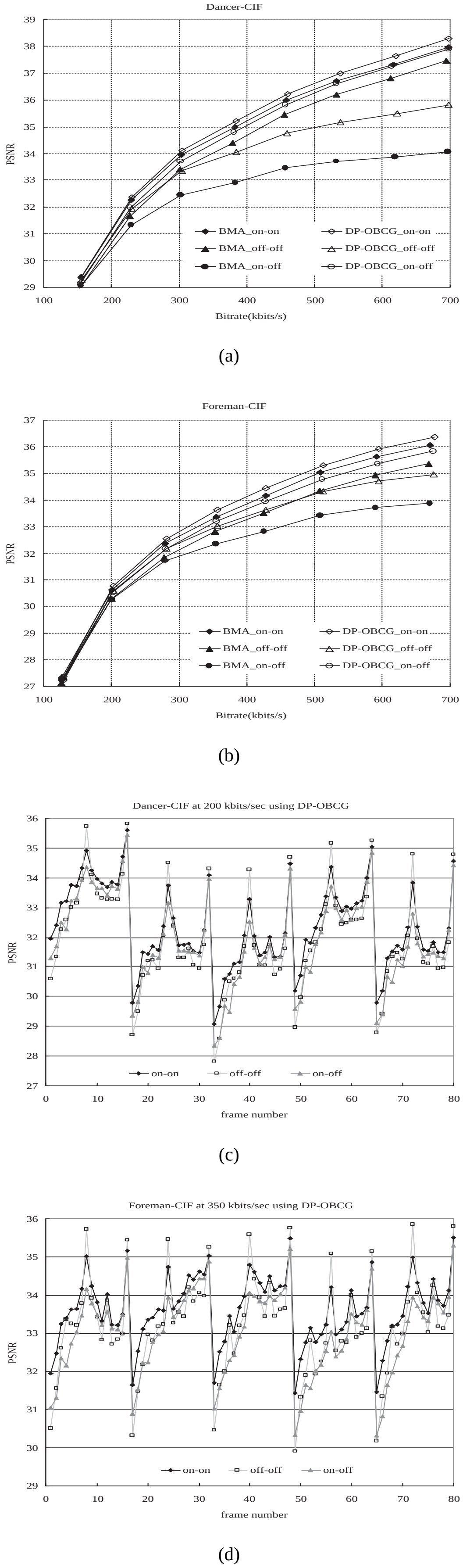
To select the proper mode of a block, either CGI or OBMC, we examine the local image characteristics. When the motion activity of a block is relatively high, CGI is used because it has a stronger smoothing operation that produces a better de-blocking effect. In contrast, OBMC is sufficient to achieve the desired de-blocking effect for blocks in which the motion activity is relatively low. The mode decision of a block is based on the following motion activity measurement:

$$\phi_{i,j} = \begin{cases} 1 & , if \max\{|\mathbf{v}_{i+p,j+q} - \mathbf{v}_{i,j}| \mid p,q = -1,0,1\} \\ & \geq Threshold \\ 0 & , otherwise. \end{cases} \tag{10}$$

In (10), if the difference between the target block's MV ($\mathbf{v}_{i,j}$) and any of its neighboring block's MV ($\mathbf{v}_{i+p,j+q}$) is larger than a pre-defined threshold, then $\phi_{i,j}$ is set to 1 and CGI is adopted for this block. Otherwise, $\phi_{i,j}$ is set to 0, and OBMC is adopted. Our goal is to find the optimal MVs that minimize

$$\begin{aligned} \sum_{i,j} [C^{OBMC}(P_{i,j}, V^{OBCG}(P_{i,j})) \cdot (1 - \phi_{i,j}) \\ + C^{CGI}(P_{i,j}, V^{OBCG}(P_{i,j})) \cdot \phi_{i,j}], \end{aligned} \tag{11}$$

where $V^{OBCG}(P_{i,j}) = \{\mathbf{v}_{i+p,j+q} \mid p = -1,0,1; q = -1,0,1\}$. Our proposed OBCG assigns a single MV to each block. Therefore, CGI, which assigns four MVs to each block, must be modified to suit the configuration of OBCG. The implementation of our OBCG, shown in Figure 5, comprises two examples, (a) and (b), which represent a block compensated for by CGI and OBMC respectively. (11) can be solved by modifying the proposed algorithm (i.e., adding an extra step to calculate $\phi_{i,j}$ before executing **Steps 2 and 3** in our algorithm). The order of the computational complexity is the same as that of DP-OBMC and DP-CGI. Since the MVs are transmitted to the decoder losslessly, the decoder can use the same set of MVs to derive $\{\phi_{i,j}\}$; therefore, our system does not need to perform the proposed algorithm at the decoder, nor to transmit the mode of each block as additional overhead to the decoder.



(a)

(b)

(c)

(d)

Fig. 6. (a) and (b) The PSNRs of different update processes at various bit-rates (including MV rates). (c) and (d) The frame-by-frame PSNR of different update processes at low bit-rates.

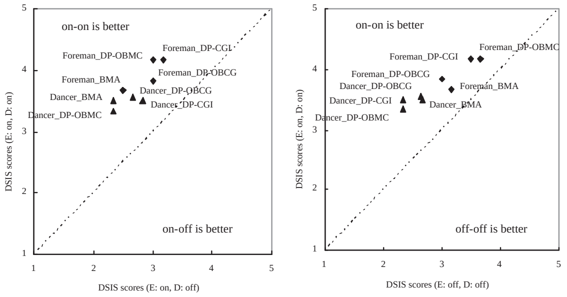


Fig. 7. Perceptual comparison of different update processes.

B. Performance Evaluation of the Hybrid System

In the following experiments, the $Threshold$ in (10) is empirically determined as 20 for CIF and 10 for QCIF sequences. The video parameters of the subjective test are set to the same values as those in Figure 3. In [15], it is reported that disabling the update step within the MCTF at the decoder reduces the PSNR, but does not affect the subjective quality of video sequences at high bit rates. We perform subjective and objective comparisons to study the effect of the update step within the MCTF at low bit rates. We use *on-on* to indicate that the update step is on at both the encoder and the decoder; *on-off* for on at the encoder, but off at the decoder; and *off-off* for off at both the encoder and the decoder. As shown in Figure 6(a), the PSNR of *on-on* is the highest; while that of *on-off* is the lowest, due to drifting. Note that we only show the results of BMA and DP-OBCG, as our experiments on OBMC and CGI yielded the same results. Their performance is not shown so that the figure is easier to read. Figure 6(b) compares the PSNR frame-by-frame, and shows that *on-on* has the smallest PSNR fluctuation, which is consistent with the result in [15]. Figure 7 shows that *on-on* achieves a better subjective result than *on-off* or *off-off*. We conclude that, both subjectively and objectively, *on-on* outperforms *on-off* and *off-off* at low bit rates for all the motion compensation methods, i.e., BMA, DP-OBMC, DP-CGI, and DP-OBCG. Hence, the following comparisons are made in the case of *on-on*.

Figures 8 and 9(a) compare the performance of BM-OBCG and DP-OBCG. We conclude that the proposed motion estimation algorithm improves both the PSNR and the visual quality of BM-OBCG. We also evaluate DP-OBCG by comparing its performance with the coding schemes of DP-CGI and DP-OBMC. Figure 8 shows the PSNR of different video sequences under various bit-rates. Because DP-CGI and DP-OBMC are special cases of DP-OBCG, the PSNR of DP-OBCG lies between the two methods. One can observe that the PSNRs of DP-OBCG are slightly lower than those of DP-OBMC; however, they are significantly better than those of DP-CGI. The subjective comparisons of DP-CGI, DP-OBMC, and DP-OBCG, shown in Figures 9(b) and 9(c), demonstrate that the visual quality of DP-OBCG sequences is substantially better than those of DP-CGI and DP-OBMC. Figure 10 compares several snapshots of video sequences encoded at low bit-rates. We also used another subjective test in which the testers were asked to identify, frame-by-frame, the blocks containing the

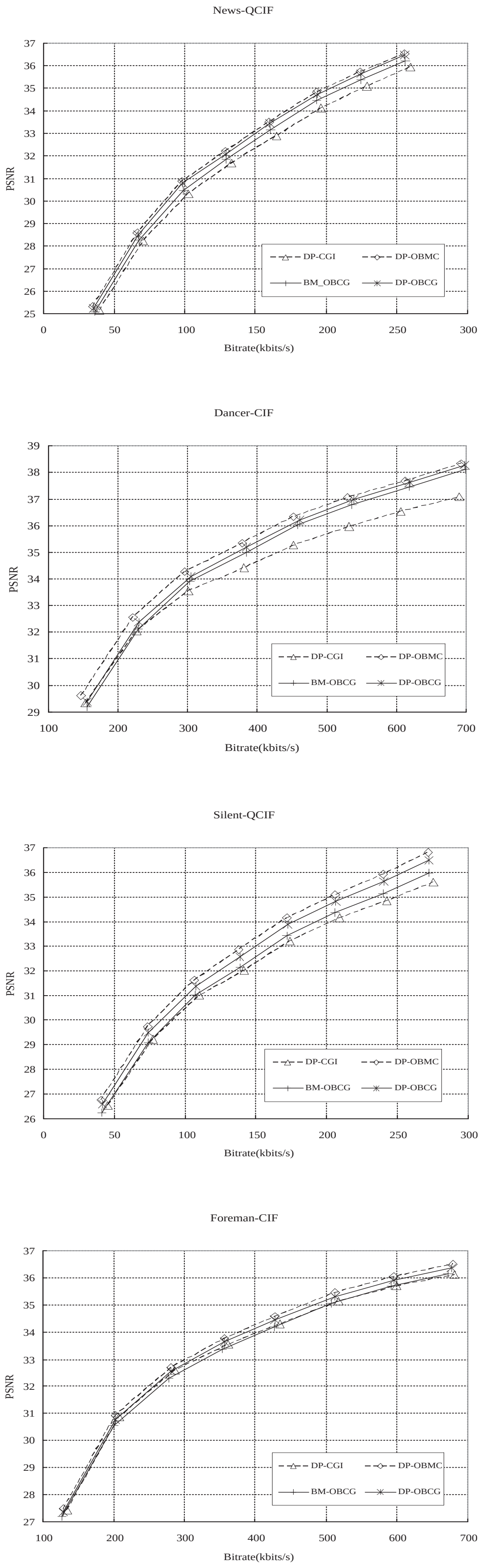


Fig. 8. The PSNRs of DP-CGI, DP-OBMC, BM-OBCG, and DP-OBCG at various bit-rates (including MV rates) in different formats for various video sequences.

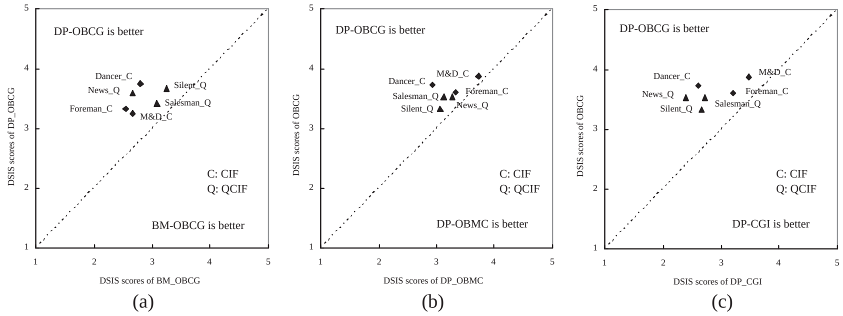
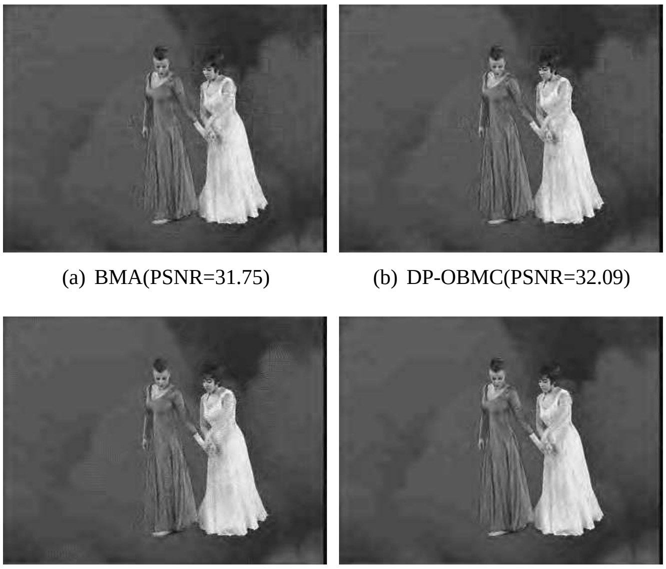


Fig. 9. Perceptual comparison of BM-OBCG, DP-OBCG, DP-CGI, and DP-OBMC.



(a) BMA(PSNR=31.75)

(b) DP-OBMC(PSNR=32.09)

(c) DP-CGI(PSNR=31.63)

(d) DP-OBCG(PSNR=32.07)

Fig. 10. Perceptual quality of Frame 40 of the *Dancer* at 230 kbits/sec using various coding techniques.

blocking artifacts in video sequences encoded by different methods. From the results shown in Figure 4(b), we conclude that OBCG removes more than 50% of the blocking artifacts that appear in OBMC sequences.

IV. Conclusion

We have compared, both objectively and subjectively, the performance of various advanced motion compensation methods designed to reduce blocking artifacts in a 3-D wavelet-based coding system. The optimal motion vectors were obtained by applying a sequence of 1-D dynamic programming algorithms. The experiment results show that our proposed hybrid OBMC/CGI system combines the high PSNR of OBMC sequences and the low blocking artifacts of CGI sequences. In our future research, we will improve the proposed system by adding more sophisticated functions. This will allow us to compare our approach with mature codecs.

Acknowledgment

The authors wish to thank the reviewers for many valuable comments and suggestions, which helped us improve the manuscript.

References

[1] M.C. Chen and A.N. Willson, Jr., "Motion-vector optimization of control grid interpolation and overlapped block motion compensation using iterated dynamic programming," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1145-1157, July 2000.

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

[3] K. Hanke, T. Rusert, and J.-R. Ohm, "Motion-compensated 3D video coding using smooth transitions," *Proc. SPIE Visual Communications Image Processing*, vol. 5022, pp. 933-940, 2003.

[4] G. Heising, D. Marpe, H.L. Cycon, and A.P. Petukhov, "Wavelet-based very low bit-rate video coding using image warping and overlapped block motion compensation," *Proc. IEE Vision, Image and Signal Processing*, vol. 148, no. 2, pp. 93-101, April 2001.

[5] P. Ishwar and P. Moulin, "Switched control grid interpolation for motion compensated video coding," *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 650-653, 1997.

[6] ITU-R, "ITU-R Recommendation BT.500-11: Methodology for the subjective assessment of the quality of television pictures," *The International Telecommunication Union*, 2002.

[7] B.-J. Kim, Z. Xiong, and W.A. Pearlman, "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 8, pp. 1374-1387, December 2000.

[8] S.D. Kim, J. Yi, H.M. Kim, and J.B. Ra, "A deblocking filter with two separate modes in block-based video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 1, pp. 156-160, February 1999.

[9] T.-Y. Kuo and C.-C.J. Kuo, "Fast overlapped block motion compensation with checkerboard block partitioning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 6, pp. 705-712, October 1998.

[10] J.-R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 559-571, September 1994.

[11] J.-R. Ohm, "Motion-compensated 3-D subband coding with multiresolution representation of motion parameters," *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 250-254, 1994.

[12] A. Secker and D. Taubman, "Highly scalable video compression using a lifting-based 3D wavelet transform with deformable mesh motion compensation," *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 749-752, 2002.

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

[14] G.J. Sullivan and R.L. Baker, "Motion compensation for video compression using control grid interpolation," *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 4, pp. 2713-2716, 1991.

[15] A. Tabatabai, Z. Visharam, and T. Suzuki, "Study of effect of update step in MCTF," *ISO/IEC JTC1/SC29/WG11*, JVT doc., Q026, Nice, October 2005.

[16] H. Watanabe and S. Singhal, "Windowed motion compensation," *Proc. SPIE Visual Communications Image Processing*, vol. 1605, pp. 582-589, 1991.

[17] Y. Wu, R.A. Cohen, and J.W. Woods, "An overlapped block motion estimation for MC-EZBC," *ISO/IEC JTC1/SC29/WG11*, MPEG doc., M10158, Brisbane, October 2003.