DCT-Based Motion Estimation

Ut-Va Koc, *Member, IEEE,* and K. J. Ray Liu, *Senior Member, IEEE*

***Abstract*—We propose novel discrete cosine transform (DCT) pseudophase techniques to estimate shift/delay between two one-dimensional (1-D) signals directly from their DCT coefficients by computing the pseudophase shift hidden in DCT and then employing the sinusoidal orthogonal principles, applicable to signal delay estimation remote sensing. Under the two-dimensional (2-D) translational motion model, we further extend the pseudophase techniques to the DCT-based motion estimation (DXT-ME) algorithm for 2-D signals/images. The DXT-ME algorithm has certain advantages over the commonly used full search block-matching approach (BKM-ME) for application to video coding despite certain limitations. In addition to its robustness in a noisy environment and low computational complexity, $O(M^2)$ for an $M \times M$ search range in comparison to the $O(N^2 \cdot M^2)$ complexity of BKM-ME for an $N \times N$ block, its ability to estimate motion completely in DCT domain makes possible the fully DCT-based motion-compensated video coder structure, which has only one major component in the feedback loop instead of three as in the conventional hybrid video coder design, and thus results in a higher system throughput. Furthermore, combination of the DCT and motion estimation units can provide space for further optimization of the overall coder. In addition, the DXT-ME algorithm has solely highly parallel local operations and this property makes feasible parallel implementation suitable for very large scale integration (VLSI) design. Simulation on a number of video sequences is presented with comparison to BKM-ME and other fast block search algorithms for video coding applications even though DXT-ME is completely different from any block search algorithms.**

***Index Terms*— Discrete cosine transform, motion estimation, shift measurement, time delay estimation, video coding.**

I. Introduction

In RECENT years, there has been great interest in motion estimation from two two-dimensional (2-D) signals or a sequence of images due to its various promising areas [1] in applications such as computer vision, image registration, target tracking, video coding with application to high definition television (HDTV), multimedia, and video telephony. Extensive research has been done over many years in developing new algorithms [1], [2] and designing cost-effective and massively parallel hardware architectures [3]–[6] suitable for current very large scale integration (VLSI) technology. Similar interests are also found in estimation of shift for the case of one-dimensional (1-D) signals, a common problem in many areas of signal processing such as time delay estimation [7], [8] and optical displacement measurement [9]. As a matter of fact, shift estimation for 1-D signals and translational motion estimation for 2-D images inherently address the same problem and can use similar techniques to approach.

In video coding, the most commonly used motion estimation scheme is the full search block-matching algorithm (BKM-ME), which searches for the best candidate block among all the blocks in a search area of larger size in terms of either the mean-square error [10] or the mean of the absolute frame difference [11]. The computational complexity of this approach is very high, i.e., $O(N^2 \cdot M^2)$ for an $N \times N$ block in an $M \times M$ search range. Even so, BKM-ME has been successfully implemented on VLSI chips [3]–[5]. To reduce the number of computations, a number of suboptimal fast block-matching algorithms have been proposed [10]–[15]. However, these algorithms require three or more sequential steps to find suboptimal estimates. Recently, a correlation-based approach [16] using complex lapped transform (CLT-ME) to avoid the blocking effect was proposed, but it still requires searching over a larger search area and thus results in a very high computational burden. Moreover, motion estimation using the CLT-ME is accurate on moving sharp edges but not on blur edges.

In addition to block-based approaches, pel-based estimation methods such as pel-recursive algorithm (PRA-ME) [17], [18] and optical flow approach (OFA-ME) [19], are very vulnerable to noise by virtue of their involving only local operations and may suffer from the instability problem.

In the category of transform-domain motion estimation algorithms, the fast Fourier transform (FFT) phase correlation method was first proposed by Kuglin and Hines [20] and then further investigated by Thomas [21] and Girod [22]. This FFT approach utilizes correlation of FFT coefficients to estimate shifts between two images from the FFT phases. However, the FFT operates on complex numbers and is not used in most video standards. Furthermore, correlation multiplies any distortion already present in the FFT of signals. For multiframe motion detection, three-dimensional (3-D) FFT has been successfully used to estimate motion in several consecutive frames [23], [24] based on the phenomenon that the spatial and temporal frequencies of a moving object lie on a plane of spatiotemporal space [25]. This requires processing of several frames rather than two.

In this paper, we present new techniques called the *DCT pseudophase techniques* [26], [27] applicable to delay estimation for 1-D signals or motion estimation for 2-D images. Unlike other fast block search motion estimation methods (such as logarithmic, three-step search, cross, subsampled

Manuscript received April 24, 1996; revised May 14, 1997. This work was supported in part by the Office of Naval Research under Grant N00014-93-1-0566, by the National Science Foundation under Award MIP9457397, and by MIPS/MicroStar. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Janusz Konrad.

U.-V. Koc is with Lucent Technologies, Bell Laboratories, Murray Hill, NJ 07974 USA (e-mail: koc@lucent.com).

K. J. R. Liu is with the Department of Electrical Engineering and Institute for Systems Research, University of Maryland, College Park, MD 20742 USA (e-mail: kjrliu@eng.umd.edu).

Publisher Item Identifier S 1057-7149(98)04370-X.

methods, etc.), which simply pick several displacement candidates out of all possible displacement values in terms of minimum MAD values of a reduced number of pixels, the new techniques employ the sinusoidal orthogonal principles to extract shift information from the pseudophases hidden in the discrete cosine transform (DCT) coefficients of signals/images.

Under the 2-D translational motion model, the techniques result in the DCT-based motion estimation (DXT-ME) algorithm, a novel algorithm for motion estimation to estimate displacements in the DCT domain. Being applied to video coding, this algorithm has certain merits over conventional methods. In addition to low computational complexity (on the order of M^2 compared to $N^2 \cdot M^2$ for BKM-ME for the search range M and block size N) and robustness of the DCT pseudophase techniques, this algorithm takes DCT coefficients of images as input to estimate motion. Therefore, it can be incorporated efficiently with the DCT-based coders used for most current video compression standards as the fully DCT-based video coder structure. It enables combining both the DCT and motion estimation into a single component to further reduce the coder complexity and at the same time increases the system throughput as explained in details in Section IV. Finally, due to the fact that the computation of pseudo phases involves only highly local operations, a highly parallel pipelined architecture for this algorithm is possible.

However, similar to other block-based transform-domain methods, the DCT-based approach suffers from the *boundary effect*, which arises from the assumption that the object moves within the block boundary. When the displacement is large compared to the block size, as a result the moving object may move partially or completely out of the block, making the contents in two temporally consecutive blocks very different. Even though this problem also exists in other motion estimation algorithms, the boundary effect becomes more severe for the DXT-ME algorithm, which enjoys lower computational complexity partly from restricting the search area to the block size than the block-matching algorithms. Therefore, the larger the block, the better its estimation. On the other hand, if the block is too large, it is difficult to use a combination of translational movements to approximate nontranslational motion as in the case of block-matching approaches. As a result, the DCT-based approach is weak at nontranslational motion estimation and good at estimation of slow motion, meaning that most of the object's features remains in the block after movement. To alleviate the boundary effect, a preprocessing step is added to remove strong background features before DCT-based motion estimation. Furthermore, for fair comparison with the full search block-matching approach (BKM-ME) having a larger search area, an adaptive overlapping approach is introduced to allow a larger search area in order to alleviate the boundary effect that occurs when displacements are large compared to the block size and the contents of two blocks differ considerably. Similar to most block-based motion estimation algorithms, the DXT-ME algorithm does not treat multiple moving objects in a block.

In the next section, we introduce the DCT pseudophase techniques with application to estimation of shift between 1-D signals. In Section III, we consider the 2-D translation motion model and extend the DCT pseudophase techniques to the DXT-ME algorithm for application to video coding. In Section IV, we discuss the various advantages of the fully DCT-based video coder architecture made possible by the DXT-ME algorithm over the conventional hybrid DCT video coder architecture. However, this paper is limited to the discussion of DCT-based motion estimation techniques, while the issue of DCT-based motion compensation is addressed in [28]. Issues related to the DCT-based video coder architecture or similar issues can be found in [29] and [30]. The preprocessing step and adaptive overlapping approach are also discussed in Section IV. Then simulation results on a number of video sequences of different characteristics are presented. Finally, the paper is concluded in Section V.

II. DCT Pseudophase Techniques

As is well known, the FT of a signal, $x(t)$ is related to FT of its shifted (or delayed if t represents time) version, $x(t-\tau)$, by this equation:

$$\mathcal{F}\{x(t-\tau)\} = e^{-j\omega\tau}\mathcal{F}\{x(t)\} \tag{1}$$

where $\mathcal{F}\{\cdot\}$ denotes Fourier transform. The phase of Fourier transform of the shifted signal contains the information about the amount of the shift τ, which can easily be extracted. However, the DCT or its counterpart, discrete sine transform (DST), do not have any phase components as usually found in the discrete Fourier transform (DFT), but DCT (or DST) coefficients of a shifted signal do also carry this shift information. To facilitate explanation of the DCT pseudophase techniques, let us first consider the case of 1-D discrete signals. Suppose that the signal $\{x_1(n);\ n \in \{0, \cdots, N-1\}\}$ is right shifted by an amount m (in our convention, a right shift means that $m > 0$) to generate another signal $\{x_2(n);\ n \in \{0, \cdots, N-1\}\}$. The values of $x_1(n)$ are all zeros outside the support region $\mathcal{S}(x_1)$. Therefore

$$x_2(n) = \begin{cases} x_1(n-m), & \text{for } n-m \in \mathcal{S}(x_1) \\ 0, & \text{elsewhere.} \end{cases} \tag{2}$$

Equation (2) implies that both signals have resemblance to each other except that the signal is shifted. It can be shown that, for $k = 1, \cdots, N-1$,

$$\begin{bmatrix} X_2^C(k) \\ X_2^S(k) \end{bmatrix} = \begin{bmatrix} Z_1^C(k) & -Z_1^S(k) \\ Z_1^S(k) & +Z_1^C(k) \end{bmatrix} \begin{bmatrix} g_m^c(k) \\ g_m^s(k) \end{bmatrix} \tag{3}$$

where $g_m^s(k) \triangleq \sin[(k\pi/N)(m + (1/2))]$ and $g_m^c(k) \triangleq \cos[(k\pi/N)(m + (1/2))]$ are called *pseudophases* analogous to phases in the FT of shifted signals. Here, X_2^S and X_2^C are DST (DST-II) and DCT (DCT-II) of the second kind of $x_2(n)$, respectively, whereas Z_1^S and Z_1^C are DST (DST-I) and DCT (DCT-I) of the first kind of $x_1(n)$, respectively, as defined as follows [31]:

$$X_2^C(k) = \frac{2}{N} C(k) \sum_{n=0}^{N-1} x_2(n) \cos\left[\frac{k\pi}{N}\,(n+0.5)\right]$$
$$k \in \{0, \cdots, N-1\} \tag{4}$$

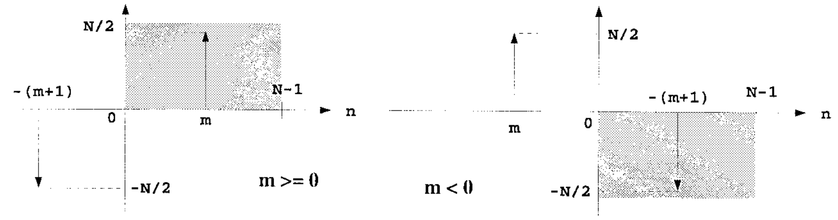


Fig. 1. Determining the direction of shift based on the sign of the peak value after application of the sinusoidal orthogonal principle for the DST-II kernel to pseudophases. (a) How to detect right shift. (b) How to detect left shift.

$$X_2^S(k) = \frac{2}{N} C(k) \sum_{n=0}^{N-1} x_2(n) \sin\left[\frac{k\pi}{N}(n+0.5)\right] \qquad k \in \{1, \cdots, N\} \tag{5}$$

$$Z_1^C(k) = \frac{2}{N} C(k) \sum_{n=0}^{N-1} x_1(n) \cos\left[\frac{k\pi}{N} n\right] \qquad k \in \{0, \cdots, N\} \tag{6}$$

$$Z_1^S(k) = \frac{2}{N} C(k) \sum_{n=0}^{N-1} x_1(n) \sin\left[\frac{k\pi}{N} n\right] \qquad k \in \{1, \cdots, N-1\} \tag{7}$$

where

$$C(k) = \begin{cases} \dfrac{1}{\sqrt{2}}, & \text{for } k = 0 \text{ or } N \\ 1, & \text{otherwise.} \end{cases}$$

As a matter of fact, the 2 × 2 matrix in (3) is orthogonal because

$$\kappa \boldsymbol{Z}_1^T(k) \boldsymbol{Z}_1(k) = \boldsymbol{I}_2 \tag{8}$$

where $\boldsymbol{I}_2$ is a 2 × 2 identity matrix, $\kappa \triangleq \{[Z_1^C(k)]^2 + [Z_1^S(k)]^2\}^{-1}$, and

$$\boldsymbol{Z}_1(k) \triangleq \begin{bmatrix} Z_1^C(k) & -Z_1^S(k) \\ Z_1^S(k) & +Z_1^C(k) \end{bmatrix}. \tag{9}$$

Therefore, it becomes very easy to solve (3) for the pseudophases $g_m^s(k)$ and $g_m^c(k)$ for $k = 1, \cdots, N-1$:

$$\vec{\boldsymbol{\theta}}(k) = \kappa \boldsymbol{Z}_1^T(k) \vec{\boldsymbol{X}}(k) \tag{10}$$

where $\vec{\boldsymbol{\theta}}(k) \triangleq [g_m^c(k), g_m^s(k)]^T$ and $\vec{\boldsymbol{X}}(k) \triangleq [X_2^C(k), X_2^S(k)]^T$. From the sinusoidal orthogonal principles

$$\frac{2}{N} \sum_{k=1}^{N} C^2(k) \sin\left[\frac{k\pi}{N}\left(m + \frac{1}{2}\right)\right] \sin\left[\frac{k\pi}{N}\left(n + \frac{1}{2}\right)\right] = \delta(m-n) - \delta(m+n+1), \tag{11}$$

$$\frac{2}{N} \sum_{k=0}^{N-1} C^2(k) \cos\left[\frac{k\pi}{N}\left(m + \frac{1}{2}\right)\right] \cos\left[\frac{k\pi}{N}\left(n + \frac{1}{2}\right)\right] = \delta(m-n) + \delta(m+n+1) \tag{12}$$

where $\delta(n)$ is the discrete impulse function, we can see that the IDST-II and IDCT-II of $g_m^s(k)$ and $g_m^c(k)$, respectively, are sums of discrete impulse functions

$$\begin{aligned} &\text{IDST-II}\{C(k) g_m^s(k)\} \\ &\quad \triangleq \frac{2}{N} \sum_{k=1}^{N} C^2(k) g_m^s(k) \sin\left[\frac{k\pi}{N}\left(n + \frac{1}{2}\right)\right] \\ &\quad = \delta(m-n) - \delta(m+n+1) \end{aligned} \tag{13}$$

$$\begin{aligned} &\text{IDCT-II}\{C(k) g_m^c(k)\} \\ &\quad \triangleq \frac{2}{N} \sum_{k=0}^{N-1} C^2(k) g_m^c(k) \cos\left[\frac{k\pi}{N}\left(n + \frac{1}{2}\right)\right] \\ &\quad = \delta(m-n) + \delta(m+n+1). \end{aligned} \tag{14}$$

The opposite signs in $\delta(m-n)$ and $\delta(m+n+1)$ of (13) are used for detecting the shift direction. If we perform an IDST-II operation on the pseudophases found, then the observable window of the index space in the inverse DST domain will be limited to $\{0, \cdots, N-1\}$. As illustrated in Fig. 1, for a right shift, one spike (generated by the positive δ function) is pointing upward at the location $n = m$ in the gray region (i.e., the observable index space), while the other δ pointing downward at $n = -(m+1)$ outside the gray region. In contrary, for a left shift, the negative spike at $n = -(m+1) > 0$ falls in the gray region but the positive δ function at $n = m$ stays out of the observable index space. It can easily be seen that a positive peak value in the gray region implies a right shift and a negative one means a left shift. This enables us to determine from the sign of the peak value the direction of the shift between signals.

The above derivation is based on assumption (2), but if the assumption is relaxed to include the noise, then

$$x_2(n) - e(n) = \begin{cases} x_1(n-m), & \text{for } n - m \in \mathcal{S}(x_1) \\ 0, & \text{elsewhere} \end{cases} \tag{15}$$

where $e(n)$ accounts for the discrepancy from the ideal assumption. Therefore

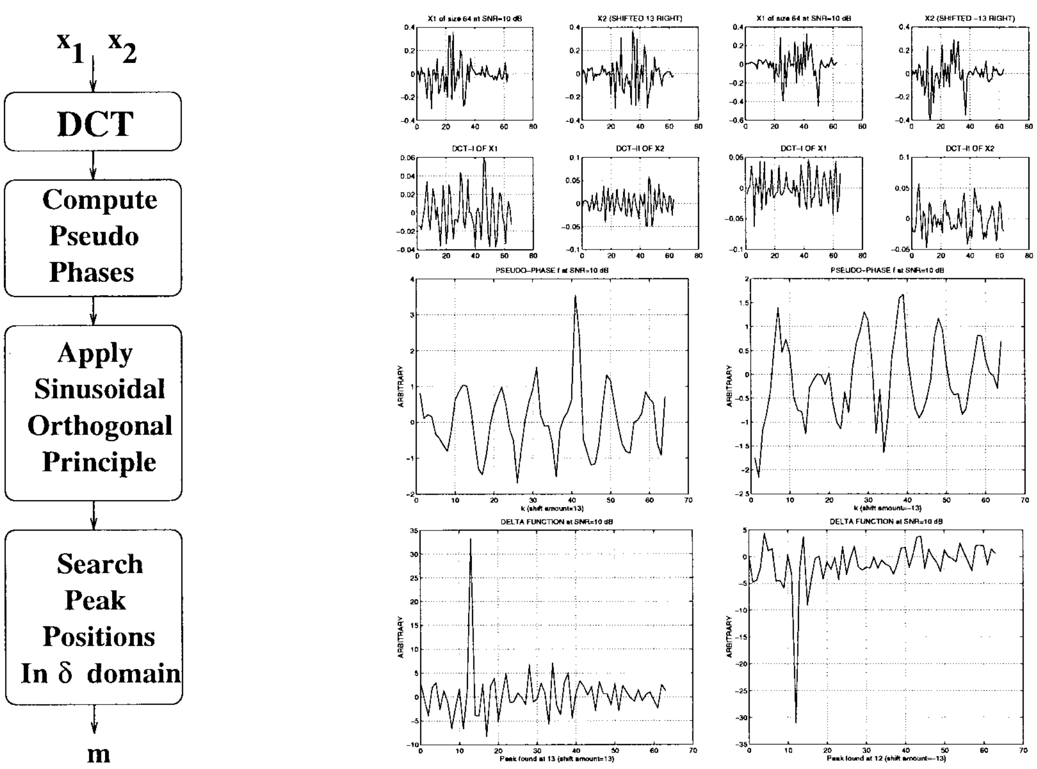
$$\vec{\boldsymbol{\theta}} = \kappa \boldsymbol{Z}^T(\vec{\boldsymbol{X}} - \vec{\boldsymbol{E}}) \tag{16}$$

where $\vec{\boldsymbol{E}}(k) = [E^C(k), E^S(k)]^T$. However, if we use (10) to compute $\vec{\boldsymbol{\theta}}$ instead because we do not know in advance $e(n)$, then the computed pseudophase values, denoted as $\vec{\hat{\boldsymbol{\theta}}}$, will be different from $\vec{\boldsymbol{\theta}}$:

$$\vec{\hat{\boldsymbol{\theta}}} = \kappa Z^T \cdot X. \tag{17}$$

Thus, the estimate error is

$$\vec{\hat{\boldsymbol{\theta}}} - \vec{\boldsymbol{\theta}} = \kappa \boldsymbol{Z}^T \cdot \vec{\boldsymbol{E}}. \tag{18}$$



(a) (b) (c)

Fig. 2. Illustration of 1-D DCT pseudophase techniques. (a) DCT pseudophase techniques. (b) Right shift. (c) Left shift.

The concept of pseudophases plus the application of the sinusoidal orthogonal principles leads to the DCT pseudophase techniques, a new approach to estimate a shift or translational motion between signals in the DCT domain as depicted in Fig. 2(a), as follows.

1) Compute the DCT-I and DST-I coefficients of $x_1(n)$ and the DCT-II and DST-II coefficients of $x_2(n)$.
2) Compute the pseudophase $\hat{g}^s_m(k)$ for $k = 1, \cdots, N$ by solving the following equation:

$$\hat{g}^s_m(k) = \begin{cases} \dfrac{Z_1^C(k) \cdot X_2^S(k) - Z_1^S(k) \cdot X_2^C(k)}{[Z_1^C(k)]^2 + [Z_1^S(k)]^2}, & \text{for } k \neq N \\ 1, & \text{for } k = N. \end{cases} \tag{19}$$

   Here, the hat notation indicates that $\hat{g}^s_m(k)$ is an estimate of the unknown real value of $g^s_m(k)$.
3) Feed the computed pseudophase, $\{C(k)\hat{g}^s_m(k);\ k = 1, \cdots, N\}$, into an IDST-II decoder to produce an output $\{d(n); n = 0, \cdots, N-1\}$, and search for the peak value. Then the estimated displacement $\hat{m}$ can be found by

$$\hat{m} = \begin{cases} i_p, & \text{if } d(i_p) > 0 \\ -(i_p + 1), & \text{if } d(i_p) < 0 \end{cases} \tag{20}$$

   where $i_p = \arg\ \max_n |d(n)|$ is the index at which the peak value is located.

In Step 1, the DCT and DST can be generated simultaneously with only $3N$ multipliers [32]–[34], and the computation of DCT-I can be easily obtained from DCT-II with minimal overhead, as will be shown later. In Step 2, if noise is absent and there is only purely translational motion, $\hat{g}_m(k)$ will be equal to $\sin(k\pi/N)(m + 0.5)$. The output $d(n)$ will then be an impulse function in the observation window. This procedure is illustrated by two examples in Fig. 2(b) and (c) with a randomly generated signal as input at signal-to-noise ratio (SNR) = 10 dB. These two examples demonstrate that the DCT pseudophase techniques are robust even in an environment of strong noise.

III. 2-D Translational Motion Model and the DXT-ME Algorithm

The DCT pseudophase techniques of extracting shift values from the pseudophases of DCT of 1-D signals, as explained in Section II, can be extended to the 2-D case. Let us confine the problem of motion estimation to this 2-D translational motion model in which an object moves translationally by m_u in X direction and m_v in Y direction as viewed on the camera plane and within the scope of a camera in a noiseless environment, as shown in Fig. 3. Then by means of the DCT pseudophase techniques, we can extract the displacement vector out of the two consecutive frames of the images of that moving object by making use of the sinusoidal orthogonal principles (11) and (12). The resulting novel algorithm for this 2-D translational motion model is called the DXT-ME algorithm, which is essentially a DCT-based motion estimation scheme.

Based on the assumption of 2-D translational displacements, we can extend the DCT pseudophase techniques to the DXT-ME algorithm depicted in Fig. 4. The previous frame x_{t-1} and the current frame x_t are fed into 2-D DCT-II and 2-D

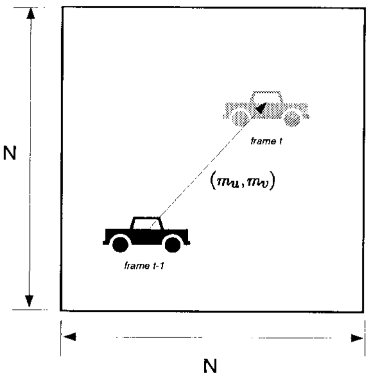


Fig. 3. An object moves translationally by m_u in X direction and m_v in Y direction as viewed on the camera plane.

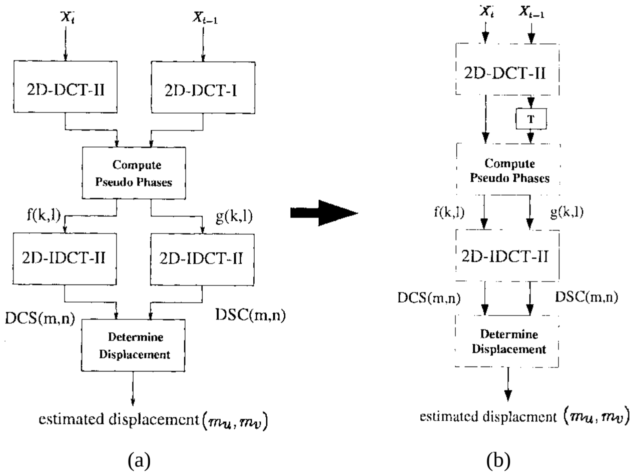


Fig. 4. Block diagram of DXT-ME (a) flowchart and (b) structure.

DCT-I, coders respectively. A 2-D DCT-II coder computes four coefficients, DCCT-II, DCST-II, DSCT-II, and DSST-II, each of which is defined as a 2-D separable function formed by 1-D DCT/DST-II kernels:

$$X_t^{cc}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_t(m,n) \cdot \cos\left[\frac{k\pi}{N}(m+0.5)\right] \cos\left[\frac{l\pi}{N}(n+0.5)\right] \quad k,l \in \{0,\cdots,N-1\} \tag{21}$$

$$X_t^{cs}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_t(m,n) \cdot \cos\left[\frac{k\pi}{N}(m+0.5)\right] \sin\left[\frac{l\pi}{N}(n+0.5)\right] \quad k \in \{0,\cdots,N-1\}, l \in \{1,\cdots,N\} \tag{22}$$

$$X_t^{sc}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_t(m,n) \cdot \sin\left[\frac{k\pi}{N}(m+0.5)\right] \cos\left[\frac{l\pi}{N}(n+0.5)\right] \quad k \in \{1,\cdots,N\}, l \in \{0,\cdots,N-1\} \tag{23}$$

$$X_t^{ss}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_t(m,n) \cdot \sin\left[\frac{k\pi}{N}(m+0.5)\right] \sin\left[\frac{l\pi}{N}(n+0.5)\right] \quad k,l \in \{1,\cdots,N\} \tag{24}$$

or symbolically

$$X_t^{cc} = \mathrm{DCCT{-}II}(x_t), \quad X_t^{cs} = \mathrm{DCST{-}II}(x_t),$$
$$X_t^{sc} = \mathrm{DSCT{-}II}(x_t), \quad X_t^{ss} = \mathrm{DSST{-}II}(x_t).$$

In the same fashion, the 2-D DCT coefficients of the first kind (2-D DCT-I) are calculated based on 1-D DCT/DST-I kernels:

$$Z_{t-1}^{cc}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m,n) \cdot \cos\left[\frac{k\pi}{N}m\right] \cos\left[\frac{l\pi}{N}n\right], \quad k,l \in \{0,\cdots,N\} \tag{25}$$

$$Z_{t-1}^{cs}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m,n) \cdot \cos\left[\frac{k\pi}{N}m\right] \sin\left[\frac{l\pi}{N}n\right], \quad k \in \{0,\cdots,N\}, l \in \{1,\cdots,N-1\} \tag{26}$$

$$Z_{t-1}^{sc}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m,n) \cdot \sin\left[\frac{k\pi}{N}m\right] \cos\left[\frac{l\pi}{N}n\right], \quad k \in \{1,\cdots,N-1\}, l \in \{0,\cdots,N\} \tag{27}$$

$$Z_{t-1}^{ss}(k,l) = \frac{4}{N^2} C(k)C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m,n) \cdot \sin\left[\frac{k\pi}{N}m\right] \sin\left[\frac{l\pi}{N}n\right], \quad k,l \in \{1,\cdots,N-1\} \tag{28}$$

or symbolically

$$Z_{t-1}^{cc} = \mathrm{DCCT{-}I}(x_{t-1}), \quad Z_{t-1}^{cs} = \mathrm{DCST{-}I}(x_{t-1}),$$
$$Z_{t-1}^{sc} = \mathrm{DSCT{-}I}(x_{t-1}), \quad Z_{t-1}^{ss} = \mathrm{DSST{-}I}(x_{t-1}).$$

Similar to the 1-D case, assuming that only translational motion is allowed, one can derive a set of equations to relate DCT coefficients of $x_{t-1}(m,n)$ with those of $x_t(m,n)$ in the same way as in (3).

$$\boldsymbol{Z}_{t-1}(k,l) \cdot \vec{\boldsymbol{\theta}}(k,l) = \vec{\boldsymbol{x}}_t(k,l), \text{for } k,l \in \mathcal{N} \tag{29}$$

where $\mathcal{N} = \{1,\cdots,N-1\}$, (30)–(32), shown at the bottom of the next page. Here $\boldsymbol{Z}_{t-1}(k,l) \in R^{4\times 4}$ is the *system matrix* of the DXT-ME algorithm at (k,l). At the boundaries of each block in the transform domain, the DCT coefficients of $x_{t-1}(m,n)$ and $x_t(m,n)$ have a 1-D relationship as given below:

$$\begin{bmatrix} Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{cs}(k,l) \\ Z_{t-1}^{cs}(k,l) & Z_{t-1}^{cc}(k,l) \end{bmatrix} \begin{bmatrix} \cos\frac{l\pi}{N}\,(m_v+0.5) \\ \sin\frac{l\pi}{N}\,(m_v+0.5) \end{bmatrix} = \begin{bmatrix} X_t^{cc}(k,l) \\ X_t^{cs}(k,l) \end{bmatrix}, \qquad k=0, l\in\mathcal{N} \tag{33}$$

$$\begin{bmatrix} Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{sc}(k,l) \\ Z_{t-1}^{sc}(k,l) & Z_{t-1}^{cc}(k,l) \end{bmatrix} \begin{bmatrix} \cos\frac{k\pi}{N}\,(m_u+0.5) \\ \sin\frac{k\pi}{N}\,(m_u+0.5) \end{bmatrix} = \begin{bmatrix} X_t^{cc}(k,l) \\ X_t^{sc}(k,l) \end{bmatrix}, \qquad l=0, k\in\mathcal{N}, \tag{34}$$

$$\begin{bmatrix} Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{cs}(k,l) \\ Z_{t-1}^{cs}(k,l) & Z_{t-1}^{cc}(k,l) \end{bmatrix} \begin{bmatrix} \cos\frac{l\pi}{N}\,(m_v+0.5) \\ \sin\frac{l\pi}{N}\,(m_v+0.5) \end{bmatrix} = (-1)^{m_u}\begin{bmatrix} X_t^{sc}(k,l) \\ X_t^{ss}(k,l) \end{bmatrix}, \qquad k=N, l\in\mathcal{N}, \tag{35}$$

$$\begin{bmatrix} Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{sc}(k,l) \\ Z_{t-1}^{sc}(k,l) & Z_{t-1}^{cc}(k,l) \end{bmatrix} \begin{bmatrix} \cos\frac{k\pi}{N}\,(m_u+0.5) \\ \sin\frac{k\pi}{N}\,(m_u+0.5) \end{bmatrix} = (-1)^{m_v}\begin{bmatrix} X_t^{cs}(k,l) \\ X_t^{ss}(k,l) \end{bmatrix}, \qquad l=N, k\in\mathcal{N} \tag{36}$$

$$(-1)^{m_v} Z_{t-1}^{cc}(k,l) = X_t^{cs}(k,l), \qquad k=0, l=N \tag{37}$$

$$(-1)^{m_u} Z_{t-1}^{cc}(k,l) = X_t^{sc}(k,l), \qquad k=N, l=0. \tag{38}$$

In a 2-D space, an object may move in four possible directions: northeast (NE: $m_u > 0$, $m_v > 0$), northwest (NW: $m_u < 0$, $m_v > 0$), southeast (SE: $m_u > 0$, $m_v < 0$), and southwest (NW: $m_u < 0$, $m_v < 0$). As explained in Section II, the orthogonal equation for the DST-II kernel in (11) can be applied to the pseudophase $\hat{g}_m^s(k)$ to determine the sign of m (i.e., the direction of the shift). In order to detect the signs of both m_u and m_v (or equivalently the direction of motion), it becomes obvious from the observation in the 1-D case that it is necessary to compute the estimated pseudophases $\hat{g}_{m_u m_v}^{SC}(\cdot,\cdot)$ and $\hat{g}_{m_u m_v}^{CS}(\cdot,\cdot)$ so that the signs of m_u and m_v can be determined from $\hat{g}_{m_u m_v}^{SC}(\cdot,\cdot)$ and $\hat{g}_{m_u m_v}^{CS}(\cdot,\cdot)$, respectively. Once again, $\hat{\cdot}$ denotes an estimated value. By taking the block boundary equations (33)–(38) into consideration, we define two pseudophase functions as in (39) and (40), shown at the bottom of the page. In the computation of $f_{m_u m_v}(k,l)$ and $g_{m_u m_v}(k,l)$, if the absolute computed value is greater than 1, then this value is ill-conditioned and should be discarded and thus we should set the corresponding variable $f_{m_u m_v}(k,l)$ or $g_{m_u m_v}(k,l)$ to be zero. This ill-conditioned situation occurs when the denominator in (39)–(40) is zero or very small and sometimes even smaller than the machine precision. This deletion of ill-conditioned values is found to improve the condition of $f_{m_u m_v}(k,l)$ and $g_{m_u m_v}(k,l)$ and also the overall performance of the DXT-ME algorithm.

These two pseudophase functions pass through 2-D IDCT-II coders (IDCST-II and IDSCT-II) to generate two functions,

$$\boldsymbol{Z}_{t-1}(k,l) = \begin{bmatrix} Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{cs}(k,l) & -Z_{t-1}^{sc}(k,l) & Z_{t-1}^{ss}(k,l) \\ Z_{t-1}^{cs}(k,l) & Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{ss}(k,l) & -Z_{t-1}^{sc}(k,l) \\ Z_{t-1}^{sc}(k,l) & -Z_{t-1}^{ss}(k,l) & Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{cs}(k,l) \\ Z_{t-1}^{ss}(k,l) & Z_{t-1}^{sc}(k,l) & Z_{t-1}^{cs}(k,l) & Z_{t-1}^{cc}(k,l) \end{bmatrix} \tag{30}$$

$$\vec{\boldsymbol{\theta}}(k,l) = \begin{bmatrix} g_{m_u m_v}^{CC}(k,l) \\ g_{m_u m_v}^{CS}(k,l) \\ g_{m_u m_v}^{SC}(k,l) \\ g_{m_u m_v}^{SS}(k,l) \end{bmatrix} = \begin{bmatrix} \cos\frac{k\pi}{N}\,(m_u+0.5)\cos\frac{l\pi}{N}\,(m_v+0.5) \\ \cos\frac{k\pi}{N}\,(m_u+0.5)\sin\frac{l\pi}{N}\,(m_v+0.5) \\ \sin\frac{k\pi}{N}\,(m_u+0.5)\cos\frac{l\pi}{N}\,(m_v+0.5) \\ \sin\frac{k\pi}{N}\,(m_u+0.5)\sin\frac{l\pi}{N}\,(m_v+0.5) \end{bmatrix} \tag{31}$$

$$\vec{\boldsymbol{x}}_t(k,l) = [X_t^{cc}(k,l) \quad X_t^{cs}(k,l) \quad X_t^{sc}(k,l) \quad X_t^{ss}(k,l)]^T \tag{32}$$

$$f_{m_u m_v}(k,l) = \begin{cases} \hat{g}_{m_u m_v}^{CS}(k,l), & \text{for } k,l\in\{1,\cdots,N-1\} \\ \dfrac{Z_{t-1}^{cc}(k,l)X_t^{cs}(k,l) - Z_{t-1}^{cs}(k,l)X_t^{cc}(k,l)}{(Z_{t-1}^{cc}(k,l))^2 + (Z_{t-1}^{cs}(k,l))^2}, & \text{for } k=0, l\in\{1,\cdots,N-1\} \\ \dfrac{Z_{t-1}^{cc}(k,l)X_t^{cs}(k,l) + Z_{t-1}^{sc}(k,l)X_t^{ss}(k,l)}{(Z_{t-1}^{cc}(k,l))^2 + (Z_{t-1}^{sc}(k,l))^2}, & \text{for } l=N, k\in\{1,\cdots,N-1\} \\ \dfrac{X_t^{cs}(k,l)}{Z_{t-1}^{cc}(k,l)}, & \text{for } k=0, l=N \end{cases} \tag{39}$$

$$g_{m_u m_v}(k,l) = \begin{cases} \hat{g}_{m_u m_v}^{SC}(k,l), & \text{for } k,l\in\{1,\cdots,N-1\}, \\ \dfrac{Z_{t-1}^{cc}(k,l)X_t^{sc}(k,l) - Z_{t-1}^{sc}(k,l)X_t^{cc}(k,l)}{(Z_{t-1}^{cc}(k,l))^2 + (Z_{t-1}^{sc}(k,l))^2}, & \text{for } l=0, k\in\{1,\cdots,N-1\} \\ \dfrac{Z_{t-1}^{cc}(k,l)X_t^{sc}(k,l) + Z_{t-1}^{cs}(k,l)X_t^{ss}(k,l)}{(Z_{t-1}^{cc}(k,l))^2 + (Z_{t-1}^{cs}(k,l))^2}, & \text{for } k=N, l\in\{1,\cdots,N-1\} \\ \dfrac{X_t^{sc}(k,l)}{Z_{t-1}^{cc}(k,l)}, & \text{for } k=N, l=0 \end{cases} \tag{40}$$

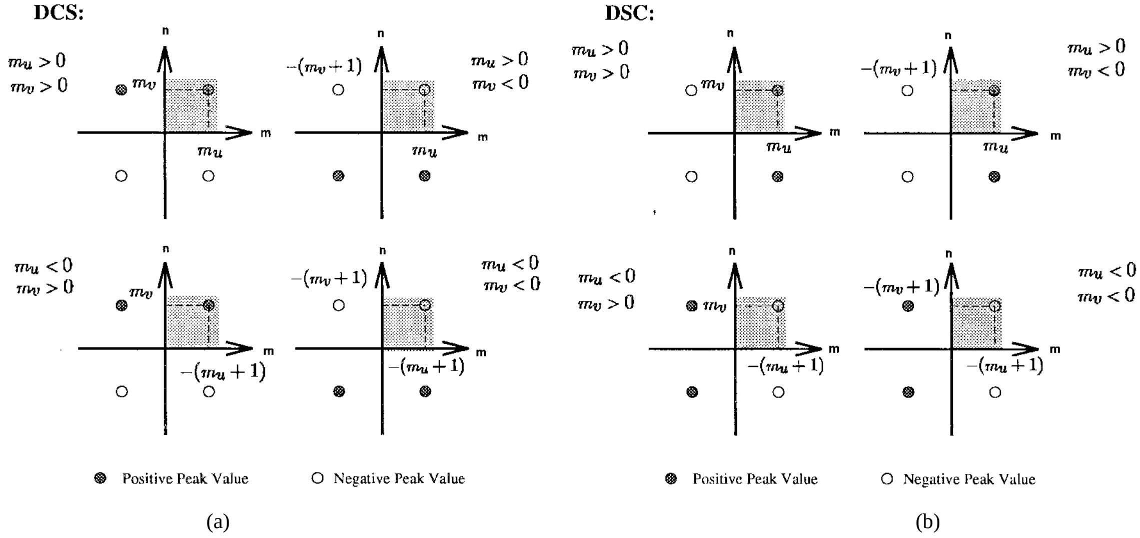


Fig. 5. Determination of the direction of motion based on the sign of the peak value. (a) DCS. (b) DSC.

DCS$(\cdot,\cdot)$ and DSC$(\cdot,\cdot)$ in view of the orthogonal property of DCT-II and DST-II in (11) and (12):

$$\begin{aligned}\mathrm{DCS}(m,n) &= \mathrm{IDCST-II}(C(k)C(l)f_{m_u m_v}(k,l))\\ &= \frac{4}{N^2}\sum_{k=0}^{N-1}\sum_{l=1}^{N} C^2(k)C^2(l)f_{m_u m_v}(k,l)\\ &\quad\cdot \cos\frac{k\pi}{N}\left(m+\frac{1}{2}\right)\sin\frac{l\pi}{N}\left(n+\frac{1}{2}\right)\\ &= [\delta(m-m_u)+\delta(m+m_u+1)]\\ &\quad\cdot[\delta(n-m_v)-\delta(n+m_v+1)] \qquad (41)\end{aligned}$$

$$\begin{aligned}\mathrm{DSC}(m,n) &= \mathrm{IDSCT-II}(C(k)C(l)g_{m_u m_v}(k,l))\\ &= \frac{4}{N^2}\sum_{k=1}^{N}\sum_{l=0}^{N-1} C^2(k)C^2(l)g_{m_u m_v}(k,l)\\ &\quad\cdot \sin\frac{k\pi}{N}\left(m+\frac{1}{2}\right)\cos\frac{l\pi}{N}\left(n+\frac{1}{2}\right)\\ &= [\delta(m-m_u)-\delta(m+m_u+1)]\\ &\quad\cdot[\delta(n-m_v)+\delta(n+m_v+1)]. \qquad (42)\end{aligned}$$

By the same argument as in the 1-D case, the 2-D IDCT-II coders limit the observable index space $\{(i,j)\colon i,j = 0,\cdots,N-1\}$ of DCS and DSC to the first quadrant of the entire index space shown as gray regions in Fig. 5, which depicts (41) and (42). Similar to the 1-D case, if m_u is positive, the observable peak value of DSC(m,n) will be positive regardless of the sign of m_v since DSC$(m,n) = \delta(m-m_u)\cdot[\delta(n-m_v)+\delta(n+m_v+1)]$ in the observable index space. Likewise, if m_u is negative, the observable peak value of DSC(m,n) will be negative because DSC$(m,n) = \delta(m+m_u+1)\cdot[\delta(n-m_v)+\delta(n+m_v+1)]$ in the gray region. As a result, the sign of the observable peak value of DSC determines the sign of m_u. The same reasoning may apply to DCS in the determination of the sign of m_v. The estimated displacement, $\hat{d} = (\hat{m}_u,\hat{m}_v)$, can thus be found by locating the peaks of DCS and DSC over $\{0,\cdots,N-1\}^2$ or over an index range of interest, usually, $\Phi = \{0,\cdots,N/2\}^2$ for slow motion. How the peak signs determine the direction of movement is summarized in Table I. Once the direction is found, $\hat{d}$ can be estimated accordingly:

TABLE I
DETERMINATION OF DIRECTION OF MOVEMENT (m_u, m_v) FROM THE SIGNS OF DSC AND DCS

Sign of DSC Peak	Sign of DCS Peak	Peak Index	Direction of Motion
+	+	(m_u, m_v)	northeast
+	−	$(m_u, -(m_v+1))$	southeast
−	+	$(-(m_u+1), m_v)$	northwest
−	−	$(-(m_u+1), -(m_v+1))$	southwest

$$\hat{m}_u = \begin{cases} i_{\mathrm{DSC}} = i_{\mathrm{DCS}}, & \text{if } \mathrm{DSC}(i_{\mathrm{DSC}}, j_{\mathrm{DSC}}) > 0\\ -(i_{\mathrm{DSC}}+1) = -(i_{\mathrm{DCS}}+1), & \\ & \text{if } \mathrm{DSC}(i_{\mathrm{DSC}}, j_{\mathrm{DSC}}) < 0\end{cases} \qquad (43)$$

$$\hat{m}_v = \begin{cases} j_{\mathrm{DCS}} = j_{\mathrm{DSC}}, & \text{if } \mathrm{DCS}(i_{\mathrm{DCS}}, j_{\mathrm{DCS}}) > 0\\ -(j_{\mathrm{DCS}}+1) = -(j_{\mathrm{DSC}}+1), & \\ & \text{if } \mathrm{DCS}(i_{\mathrm{DCS}}, j_{\mathrm{DCS}}) < 0\end{cases} \qquad (44)$$

where

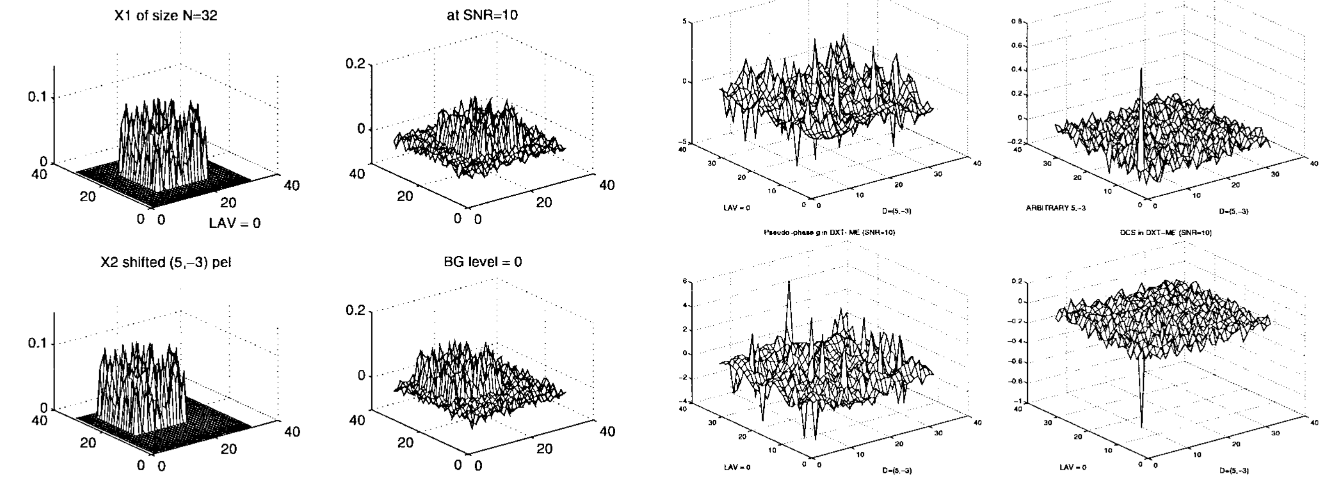
$$(i_{\mathrm{DCS}}, j_{\mathrm{DCS}}) = \arg\max_{m,n\in\Phi} |\mathrm{DCS}(m,n)| \qquad (45)$$

$$(i_{\mathrm{DSC}}, j_{\mathrm{DSC}}) = \arg\max_{m,n\in\Phi} |\mathrm{DSC}(m,n)|. \qquad (46)$$

Normally, these two peak indices are consistent but in noisy circumstances, they may not agree. In this case, an arbitration rule must be made to pick the best index (i_D, j_D) in terms of minimum nonpeak-to-peak ratio (NPR):

$$(i_D, j_D) = \begin{cases} (i_{\mathrm{DSC}}, j_{\mathrm{DSC}}) & \text{if NPR(DSC)} < \text{ NPR(DCS)}\\ (i_{\mathrm{DCS}}, j_{\mathrm{DCS}}) & \text{if NPR(DSC)} > \text{ NPR(DCS).}\end{cases} \qquad (47)$$

This index (i_D, j_D) will then be used to determine $\hat{d}$ by (43) and (44). Here, NPR is defined as the ratio of the average of all absolute nonpeak values to the absolute peak value. Thus, $0 \leq$NPR ≤ 1, and for a pure impulse function, NPR $= 0$. Such an approach to choose the best index among the two



(a) (b) (c) (d)

Fig. 6. DXT-ME performed on the images of an object moving in the direction (5, -3) with additive white Gaussian noise at SNR $=$ 10 dB in a completely dark environment. (a) Original inputs x_1 and x_2 (b) noise added.

indices is found empirically to improve the noise immunity of this estimation algorithm.

In situations where slow motion is preferred, it is better to search the peak value in a zigzag way as widely used in DCT-based hybrid video coding [35], [36]. Starting from the index (0, 0), zigzagly scan all the DCS (or DSC) values and mark the point as the new peak index if the value at that point (i,j) is larger than the current peak value by more than a preset threshold θ:

$$(i_{\text{DCS}}, j_{\text{DCS}}) = (i,j) \quad \text{if } \text{DCS}(i,j) > \text{DCS}(i_{\text{DCS}}, j_{\text{DCS}}) + \theta, \tag{48}$$

$$(i_{\text{DSC}}, j_{\text{DSC}}) = (i,j) \quad \text{if } \text{DSC}(i,j) > \text{DSC}(i_{\text{DSC}}, j_{\text{DSC}}) + \theta. \tag{49}$$

In this way, large spurious spikes at the higher index points will not affect the performance and thus improve its noise immunity further.

Fig. 6 demonstrates the DXT-ME algorithm. Images of a rectangularly-shaped moving object with arbitrary texture are generated as in Fig. 6(a) and corrupted by additive white Gaussian noise at SNR $=$ 10 dB as in Fig. 6(b). The resulted pseudophase functions f and g, as well as DCS and DSC, are depicted in Fig. 6(c) and (d), correspondingly. Large peaks can be seen clearly in Fig. 6(d) on rough surfaces caused by noise in spite of noisy input images. The positions of these peaks give us an accurate motion estimate (5, -3).

A. Analysis

What if an object is moving in a uniformly bright background instead of a completely dark environment? It can be shown analytically and empirically that uniformly bright background introduces only very small spikes which does not affect the accuracy of the estimate. Suppose that $\{x_{t-1}(m,n)\}$ and $\{x_t(m,n)\}$ are pixel values of two consecutive frames of an object displaced by (m_u, m_v) on a uniformly bright background. Then let $y_t(m,n)$ and $y_{t-1}(m,n)$ be the pixel value of $x_t(m,n)$ and $x_{t-1}(m,n)$ subtracted by the background pixel value c $(c > 0)$, respectively:

$$y_t(m,n) = x_t(m,n) - c, \tag{50}$$

$$y_{t-1}(m,n) = x_{t-1}(m,n) - c. \tag{51}$$

In this way, $\{x_{t-1}(m,n)\}$ and $\{x_t(m,n)\}$ can be considered as the images of an object moving in a dark environment. Denote $\boldsymbol{Z}_{t-1}(k,l)$ as the system matrix of the input image x_{t-1} and $\boldsymbol{U}_{t-1}(k,l)$ as that of y_{t-1} for $k,l \in \mathcal{N}$. Also let $\vec{\boldsymbol{x}}_t(k,l)$ be the vector of the 2-D DCT-II coefficients of x_t and $\vec{\boldsymbol{y}}_t(k,l)$ be the vector for y_t. Applying the DXT-ME algorithm to both situations, we have, for $k,l \in \mathcal{N}$

$$\boldsymbol{Z}_{t-1}(k,l) \cdot \vec{\boldsymbol{\theta}}_{m_u m_v}(k,l) = \vec{\boldsymbol{x}}_t(k,l) \tag{52}$$

$$\boldsymbol{U}_{t-1}(k,l) \cdot \vec{\phi}_{m_u m_v}(k,l) = \vec{\boldsymbol{y}}_t(k,l). \tag{53}$$

Here, $\vec{\phi}_{m_u m_v}(k,l)$ is the vector of the computed pseudophases for the case of dark background and thus

$$\vec{\phi}_{m_u m_v}(k,l) = [g^{CC}_{m_u m_v}(k,l), g^{CS}_{m_u m_v}(k,l), g^{SC}_{m_u m_v}(k,l) \\ g^{SS}_{m_u m_v}(k,l)]^T$$

but $\vec{\boldsymbol{\theta}}_{m_u m_v}(k,l)$ is for uniformly bright background and

$$\vec{\boldsymbol{\theta}}_{m_u m_v}(k,l) = [\hat{g}^{CC}_{m_u m_v}(k,l), \hat{g}^{CS}_{m_u m_v}(k,l), \hat{g}^{SC}_{m_u m_v}(k,l) \\ \hat{g}^{SS}_{m_u m_v}(k,l)]^T \neq \vec{\phi}_{m_u m_v}(k,l).$$

Starting from the definition of each element in $\boldsymbol{Z}_{t-1}(k,l)$ and $\vec{\boldsymbol{x}}_t(k,l)$, we obtain

$$\boldsymbol{Z}_{t-1}(k,l) = \boldsymbol{U}_{t-1}(k,l) + c \cdot \boldsymbol{D}(k,l) \tag{54}$$

$$\vec{\boldsymbol{x}}_t(k,l) = \vec{\boldsymbol{y}}_t(k,l) + c \cdot \vec{\boldsymbol{c}}(k,l) \tag{55}$$

where $\boldsymbol{D}(k,l)$ is the system matrix with $\{d(m,n) = 1, \forall m,n = \{0, \cdots, N-1\}\}$ as input and $\vec{\boldsymbol{c}}(k,l)$ is the vector of the 2D-DCT-II coefficients of $d(m,n)$. Substituting (54) and (55) into (53), we get

$$\begin{aligned}&\boldsymbol{Z}_{t-1}(k,l)\cdot\vec{\boldsymbol{\theta}}_{m_um_v}(k,l)\\&\quad=\boldsymbol{Z}_{t-1}(k,l)\cdot\vec{\phi}_{m_um_v}(k,l)+c\\&\qquad\cdot[\vec{\boldsymbol{c}}(k,l)-\boldsymbol{D}(k,l)\cdot\vec{\phi}_{m_um_v}(k,l)].\end{aligned}\tag{56}$$

Since $\vec{\boldsymbol{c}}(k,l)=\boldsymbol{D}(k,l)\cdot\vec{\phi}_{00}(k,l)$, (56) becomes

$$\begin{aligned}\vec{\boldsymbol{\theta}}_{m_um_v}(k,l)=&\,\vec{\phi}_{m_um_v}(k,l)+c\boldsymbol{Z}_{t-1}^{-1}(k,l)\boldsymbol{D}(k,l)\\&\cdot[\vec{\phi}_{00}(k,l)-\vec{\phi}_{m_um_v}(k,l)]\end{aligned}\tag{57}$$

provided that $|\boldsymbol{Z}_{t-1}(k,l)|\neq 0$. Similar results can also be found at block boundaries. Referring to (30), we know that $\boldsymbol{D}(k,l)$ is composed of $D^{cc}(k,l)$, $D^{cs}(k,l)$, $D^{sc}(k,l)$, and $D^{ss}(k,l)$, each of which is a separable function made up by

$$\begin{aligned}D^c(k)&\equiv\frac{2}{N}C(k)\sum_{m=0}^{N-1}\cos\left[\frac{k\pi}{N}m\right]\\&=\frac{2}{N}C(k)\{0.5[1-(-1)^k]+N\cdot\delta(k)\}\\D^s(k)&\equiv\frac{2}{N}C(k)\sum_{m=0}^{N-1}\sin\left[\frac{k\pi}{N}m\right]\\&=\begin{cases}\dfrac{2}{N}\,C(k)\dfrac{[1-(-1)^k]}{2\tan\dfrac{k\pi}{2N}} & \text{for } k\neq 0\\ 0 & \text{for } k=0.\end{cases}\end{aligned}$$

From the above equations, we can see that $D^c(k)=D^s(k)=0$ if k is even, and for odd $k>0$, $D^c(k)=(2/N)$ while $D^s(k)=(2/N\tan(k\pi/2N))$. Hence, $D^{cc}(k,l)=D^{cs}(k,l)=D^{sc}(k,l)=D^{ss}(k,l)=0$ if either k or l is even. As a result, $\theta_{m_um_v}(k,l)=\phi_{m_um_v}(k,l)$ if either k or l is even. For odd indices k and l, it is possible to find a constant s and a matrix $\boldsymbol{N}(k,l)\in R^{4\times 4}$ such that $\boldsymbol{U}_{t-1}(k,l)=s[\boldsymbol{D}(k,l)-\boldsymbol{N}(k,l)]$ and $|\boldsymbol{N}(k,l)\boldsymbol{D}^{-1}(k,l)|<1$ for $|\boldsymbol{D}(k,l)|\neq 0$. Therefore, for $|(s/s+c)\boldsymbol{N}(k,l)\boldsymbol{D}^{-1}(k,l)|<1$

$$\begin{aligned}&c\boldsymbol{Z}_{t-1}^{-1}(k,l)\boldsymbol{D}(k,l)\\&\quad=\frac{c}{s+c}\left[\boldsymbol{I}-\frac{s}{s+c}\,\boldsymbol{N}(k,l)\boldsymbol{D}^{-1}(k,l)\right]^{-1}\end{aligned}\tag{58}$$

$$\begin{aligned}&\quad=\frac{c}{s+c}\left\{\boldsymbol{I}+\frac{s}{s+c}\,\boldsymbol{N}(k,l)\boldsymbol{D}^{-1}(k,l)\right.\\&\qquad\left.+\left[\frac{s}{s+c}\,\boldsymbol{N}(k,l)\boldsymbol{D}^{-1}(k,l)\right]^2+\cdots\right\}.\end{aligned}\tag{59}$$

If we lump all the high-order terms of $(s/s+c)\boldsymbol{N}(k,l)\boldsymbol{D}^{-1}(k,l)$ in one term $\boldsymbol{H}(k,l)$, then

$$\begin{aligned}\vec{\boldsymbol{\theta}}_{m_um_v}(k,l)=&\,\vec{\phi}_{m_um_v}(k,l)+\left[\frac{c}{s+c}+\boldsymbol{H}(k,l)\right]\\&\cdot[\vec{\phi}_{00}(k,l)-\vec{\phi}_{m_um_v}(k,l)].\end{aligned}\tag{60}$$

Usually, $0\leq c,s\leq 255$ for the maximum gray level equal to 255. For moderately large c, $\boldsymbol{H}(k,l)$ is very small. Define the subsampled version of the pseudophase function $\vec{\phi}_{ab}(k,l)$ as

$$\vec{\lambda}_{ab}(k,l)\equiv\begin{cases}\vec{\phi}_{ab}(k,l), & \text{if both } k \text{ and } l \text{ are odd}\\ 0, & \text{otherwise.}\end{cases}\tag{61}$$

Then

$$\begin{aligned}\vec{\boldsymbol{\theta}}_{m_um_v}(k,l)=&\,\vec{\phi}_{m_um_v}(k,l)+\left[\frac{c}{s+c}+\boldsymbol{H}(k,l)\right]\\&\cdot\{\vec{\lambda}_{00}-\vec{\lambda}_{m_um_v}\}.\end{aligned}\tag{62}$$

Recall that a 2-D IDCT-II operation on $\vec{\phi}_{m_um_v}(k,l)$ or $\vec{\phi}_{00}(k,l)$ produces $\vec{\delta}_{m_um_v}$ or $\vec{\delta}_{00}$, respectively, where

$$\vec{\delta}_{ab}(m,n)=\begin{bmatrix}(\delta(m-a)+\delta(m+a+1))\\ \quad\cdot(\delta(n-b)+\delta(n+b+1))\\ (\delta(m-a)+\delta(m+a+1))\\ \quad\cdot(\delta(n-b)-\delta(n+b+1))\\ (\delta(m-a)-\delta(m+a+1))\\ \quad\cdot(\delta(n-b)+\delta(n+b+1))\\ (\delta(m-a)-\delta(m+a+1))\\ \quad\cdot(\delta(n-b)-\delta(n+b+1))\end{bmatrix}.$$

Therefore

$$\begin{aligned}\vec{\boldsymbol{d}}(m,n)\equiv&\,\text{2-D-IDCT-II}\{C(k)C(l)\vec{\boldsymbol{\theta}}_{m_um_v}(k,l)\}\\=&\,\vec{\delta}_{m_um_v}(m,n)+\frac{c}{s+c}\text{ 2-D IDCT-II}\{C(k)C(l)\\&\cdot[\vec{\lambda}_{00}(k,l)-\vec{\lambda}_{m_um_v}(k,l)]\}+\vec{\boldsymbol{n}}(m,n)\end{aligned}\tag{63}$$

where $\vec{\boldsymbol{n}}$ is the noise term contributed from 2-D IDCT-II$\{\boldsymbol{H}(k,l)C(k)C(l)[\vec{\lambda}_{00}(k,l)-\vec{\lambda}_{m_um_v}(k,l)]\}$. Because $\vec{\lambda}_{ab}$ is equivalent to downsampling $\vec{\phi}_{ab}$ in a 2-D index space and it is known that downsampling produces in the transform domain mirror images of magnitude only one-fourth of the original and of sign depending on the transform function, we obtain

$$\begin{aligned}\vec{\boldsymbol{E}}_{m_um_v}(m,n)\equiv&\,\text{2-D IDCT-II}\{C(k)C(l)\vec{\lambda}_{m_um_v}(k,l)\}\\=&\,\tfrac{1}{4}\,[\vec{\delta}_{m_um_v}(m,n)\\&+\text{diag}(\vec{\boldsymbol{\zeta}}_1)\cdot\vec{\delta}_{(N-1-m_u)m_v}(m,n)\\&+\text{diag}(\vec{\boldsymbol{\zeta}}_2)\cdot\vec{\delta}_{m_u(N-1-m_v)}(m,n)\\&+\text{diag}(\vec{\boldsymbol{\zeta}}_3)\cdot\vec{\delta}_{(N-1-m_u)(N-1-m_v)}(m,n)]\end{aligned}\tag{64}$$

where diag$(\cdot)$ is the diagonal matrix of a vector and $\vec{\boldsymbol{\zeta}}_i$ $(i=1,2,3)$ is a vector consisting of ±1. A similar expression can also be established for 2–D DCT-II$\{\vec{\lambda}_{00}\}$. In conclusion

$$\begin{aligned}\vec{\boldsymbol{d}}(m,n)=&\,\vec{\delta}_{m_um_v}(m,n)+\frac{c}{4(s+c)}\\&\cdot[\vec{\boldsymbol{E}}_{00}(m,n)-\vec{\boldsymbol{E}}_{m_um_v}(m,n)]+\vec{\boldsymbol{n}}(m,n).\end{aligned}\tag{65}$$

The above equation predicts the presence of a very small noise term $\vec{\boldsymbol{n}}$ and several small spikes, $\vec{\boldsymbol{E}}_{00}$ and $\vec{\boldsymbol{E}}_{m_um_v}$, of magnitude moderated by $(c/4(s+c))$ which are much smaller than the displacement peak, as displayed in Fig. 7(b) and (c) where $\vec{\boldsymbol{n}}$ for the case of $c=3$ in (b) is observable but very small, and can be regarded as noise whereas $\vec{\boldsymbol{n}}$ is practically absent as in (c) when $c=255$.

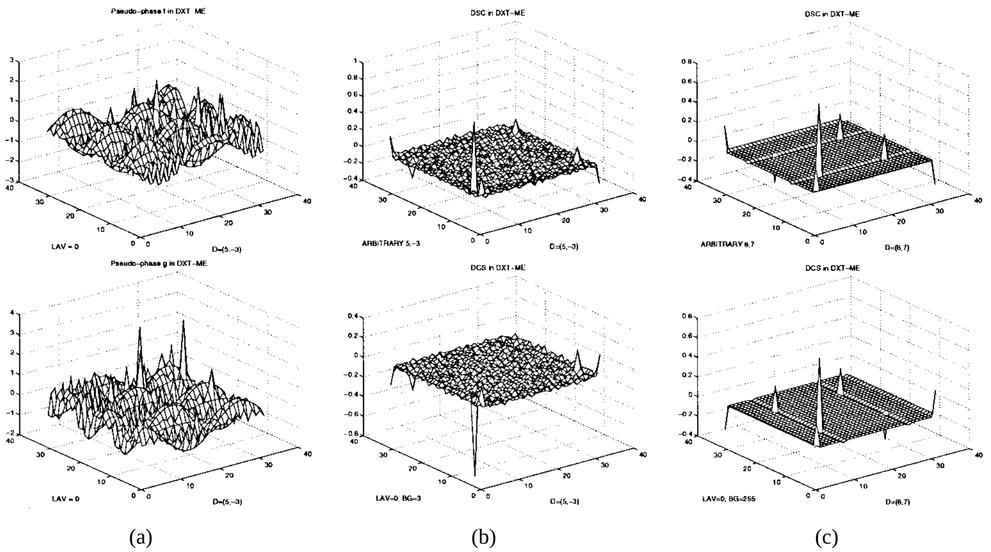


Fig. 7. (a), (b) An object is moving in the direction (5, −3) in a uniformly bright background $(c = 3)$. (c) Another object is moving northeast (8, 7) for background pixel values $= c = 255$. (a) f and g. (b) DSC and DCS. (c) Another DSC and DCS.

B. Computational Issues and Complexity

The block diagram in Fig. 4(a) shows that a separate 2-D DCT-I is needed in addition to the standard DCT (2-D DCT-II). This is undesirable from the complexity viewpoint. However, this problem can be circumvented by considering the point-to-point relationship between 2-D DCT-I and 2-D DCT-II coefficients in the frequency domain for $k, l \in \mathcal{N}$, as in (66), shown at the bottom of the page, where X_{t-1}^{cc}, X_{t-1}^{cs}, X_{t-1}^{sc}, and X_{t-1}^{ss} are the 2-D DCT-II coefficients of the previous frame. Similar relation also exists for the coefficients at block boundaries. This observation results in the simple structure in Fig. 4(b), where Block T is a coefficient transformation unit realizing (66).

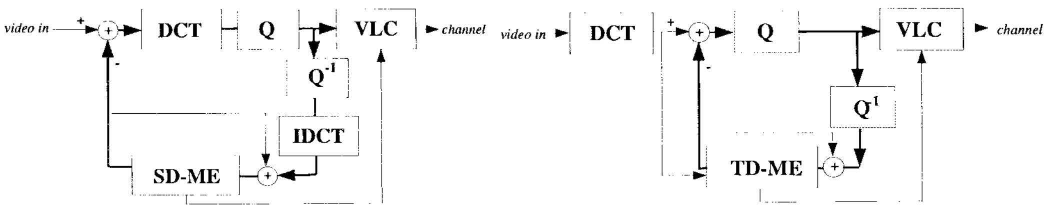
In view of the fact that the actual number of computations required by the DCT pseudophase techniques or the DXT-ME algorithm lies heavily on the specific implementation for a particular application such as motion estimation in video coding, it is more appropriate to consider the asymptotic computational complexity as generally accepted in the evaluation of algorithms in this section. Based on the straightforward implementation without further optimization, a rough count of the actual number of computations will be presented in Section IV where the DXT-ME algorithm is used in video coding.

If the DCT has computational complexity O_{dct}, the overall complexity of DXT-ME is $O(M^2)+O_{dct}$ with the complexity of each component summarized in Table II. The computational complexity of the pseudophase computation component is only $O(M^2)$ for an $M \times M$ search range and so is the unit to determine the displacement. For the computation of the pseudophase functions $f(\cdot,\cdot)$ in (39) and $g(\cdot,\cdot)$ in (40), DSCT, DCST, and DSST coefficients (regarded as DST coefficients) must be calculated in addition to DCCT coefficients (i.e., the usual 2-D DCT). However, all these coefficients can be generated with little overhead in the course of computing 2-D DCT coefficients. As a matter of fact, a parallel and fully-pipelined 2-D DCT lattice structure has been developed [32]–[34] to generate 2-D DCT coefficients at a cost of $O(M)$ operations. This DCT coder computes DCT and DST coefficients dually due to its internal lattice architecture. These internally generated DST coefficients can be output to the DXT-ME module for pseudophase computation. This same lattice structure can also be modified as a 2-D IDCT which also has $O(M)$ complexity. To sum up, the computational complexity of this DXT-ME is only $O(M^2)$, lower than the $O(N^2 \cdot M^2)$ complexity of BKM-ME for an $N \times N$ block.

TABLE II
COMPUTATIONAL COMPLEXITY OF EACH STAGE IN DXT-ME FOR A SEARCH RANGE $M \times M$

Stage	Component	Computational Complexity
1	2D-DCT-II	$O_{dct} = O(M)$
	Coeff. Transformation Unit (T)	$O(M^2)$
2	Pseudo Phase Computation	$O(M^2)$
3	2D-IDCT-II	$O_{dct} = O(M)$
4	Peak Searching	$O(M^2)$
	Estimation	$O(1)$

$$\begin{bmatrix} Z_{t-1}^{cc}(k,l) \\ Z_{t-1}^{cs}(k,l) \\ Z_{t-1}^{sc}(k,l) \\ Z_{t-1}^{ss}(k,l) \end{bmatrix} = \begin{bmatrix} \cos\frac{k\pi}{2N}\cos\frac{l\pi}{2N} & \cos\frac{k\pi}{2N}\sin\frac{l\pi}{2N} & \sin\frac{k\pi}{2N}\cos\frac{l\pi}{2N} & \sin\frac{k\pi}{2N}\sin\frac{l\pi}{2N} \\ -\cos\frac{k\pi}{2N}\sin\frac{l\pi}{2N} & \cos\frac{k\pi}{2N}\cos\frac{l\pi}{2N} & -\sin\frac{k\pi}{2N}\sin\frac{l\pi}{2N} & \sin\frac{k\pi}{2N}\cos\frac{l\pi}{2N} \\ -\sin\frac{k\pi}{2N}\cos\frac{l\pi}{2N} & -\sin\frac{k\pi}{2N}\sin\frac{l\pi}{2N} & \cos\frac{k\pi}{2N}\cos\frac{l\pi}{2N} & \cos\frac{k\pi}{2N}\sin\frac{l\pi}{2N} \\ \sin\frac{k\pi}{2N}\sin\frac{l\pi}{2N} & -\sin\frac{k\pi}{2N}\cos\frac{l\pi}{2N} & -\cos\frac{k\pi}{2N}\sin\frac{l\pi}{2N} & \cos\frac{k\pi}{2N}\cos\frac{l\pi}{2N} \end{bmatrix} \begin{bmatrix} X_{t-1}^{cc}(k,l) \\ X_{t-1}^{cs}(k,l) \\ X_{t-1}^{sc}(k,l) \\ X_{t-1}^{ss}(k,l) \end{bmatrix} \tag{66}$$



(a)

(b)

Fig. 8. Coder structures: (a) Conventional hybrid DCT motion-compensated video coder. (b) Fully DCT-based motion-compensated video coder.

A closer look at (39), (40), and (66) reveals that the operations of pseudophase computation and coefficient transformation are performed independently at each point (k, l) in the transform domain and therefore are inherently highly parallel operations. Since most of the operations in the DXT-ME algorithm involve mainly pseudophase computation and coefficient transformation in addition to DCT and IDCT operations which have been studied extensively, the DXT-ME algorithm can easily be implemented on highly parallel array processors or dedicated circuits. This is very different from BKM-ME, which requires shifting of pixels and summation of differences of pixel values and, hence, discourages parallel implementation.

IV. Application to Video Coding and Fully DCT-Based Video Coder

It is because the proposed DCT pseudophase techniques and the DXT-ME algorithm are DCT-based that the immediate application of the algorithm will then be motion estimation incorporated into the standard-compliant DCT-based motion-compensated video coder design. In most international video coding standards such as CCITT H.261 [35], MPEG [36] as well as the proposed HDTV standard, DCT- and block-based motion estimation are the essential elements to achieve spatial and temporal compression, respectively. Most implementations of a standard-compliant coder adopt the conventional motion compensated DCT video coder structure as shown in Fig. 8(a). The DCT is located inside the loop of temporal prediction, which also includes an IDCT and a spatial-domain motion estimator (SD-ME), which is usually the BKM-ME. The IDCT is needed solely for transforming the DCT coefficients back to the spatial domain in which the SD-ME estimates motion vectors and performs motion compensated prediction. This is an undesirable coder architecture for the following reasons. In addition to the additional complexity added to the overall architecture, the DCT and IDCT must be put inside the feedback loop, which has long been recognized as the major bottleneck of the entire digital video system for high-end real-time applications. The throughput of the coder is limited by the processing speed of the feedback loop, which is roughly the total time for the data stream to go through each component in the loop. Therefore the DCT (or IDCT) must be designed to operate at least twice as fast as the incoming data stream. A compromise is to remove the loop and perform open-loop motion estimation based upon original images instead of reconstructed images in sacrifice of the performance of the coder [37].

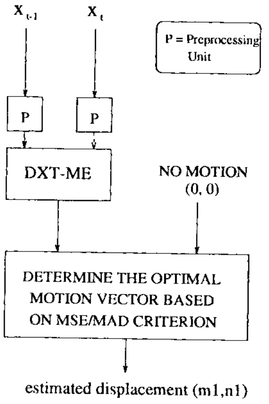

X_{t-1}
X_t
P = Preprocessing Unit
P
P
DXT-ME
NO MOTION (0, 0)
DETERMINE THE OPTIMAL MOTION VECTOR BASED ON MSE/MAD CRITERION
estimated displacement (m1,n1)

Fig. 9. Block diagram of simplified extended DXT-ME.

An alternative solution without degradation of the performance is to develop a motion estimation algorithm that can work in the DCT transform domain as remarked in [38]. In this way, the DCT can be moved out of the loop as depicted in Fig. 8(b), and thus the operating speed of this DCT can be reduced to the data rate of the incoming stream. Moreover, the IDCT is removed from the feedback loop, which now has only two simple components Q and Q^{-1} (the quantizers) in addition to the transform-domain motion estimator (TD-ME). This not only reduces the complexity of the coder but also resolve the bottleneck problem with little tradeoff of the performance. In fact, the benefit of lower overall complexity comes largely from the combined DCT and motion estimation operation. Furthermore, as pointed out in [38], different components can be jointly optimized if they operate in the same transform domain. It should be stressed that by using the DXT-ME algorithm discussed in this paper and the DCT-based motion compensation methods investigated in [28]–[30], standard-compliant bitstreams can be formed in accordance to the specification of any DCT-based standard such as MPEG without any need to change the structure of any standard-compliant decoder. This standard compliance implies an architecturally change to improve the MPEG encoder speed at a reduced cost.

A. Preprocessing

For complicated video sequences in which objects may move across the border of blocks in nonuniform background, preprocessing can be employed to enhance the features of moving objects and avoid violation of the assumption of DXT-ME that the only moving object moves completely

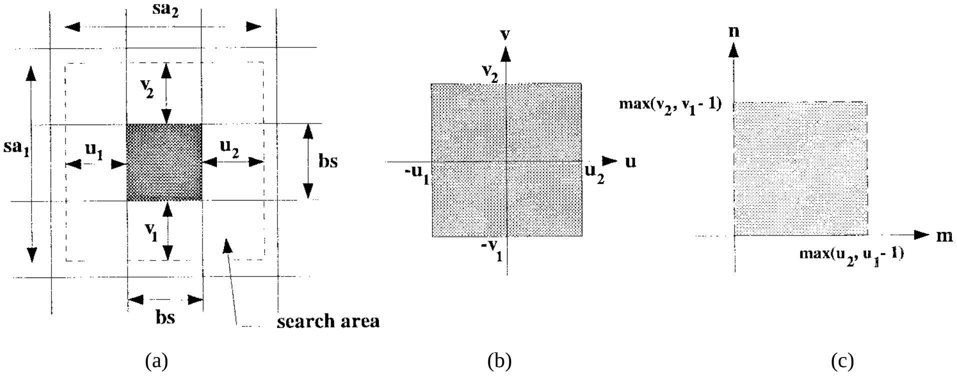


Fig. 10. Adaptive overlapping approach.

inside the block boundary. Intuitively speaking, the DXT-ME algorithm tries to match the features of any object on two consecutive frames so that any translation motion can be estimated regardless of the shape and texture of the object as long as these two frames contain significant energy level of the object features. Due to this feature matching property of the DXT-ME algorithm, effective preprocessing will improve the performance of motion estimation if preprocessing can enhance the object features in the original sequence. In order to keep the computational complexity of the overall motion estimator low, the chosen preprocessing function must be simple but effective in the sense that unwanted features will not affect the accuracy of estimation. Our study found that both edge extraction and frame differentiation are simple and effective schemes for extraction of motion information.

Edges of an object can represent the object itself in motion estimation as its features [39] and contain the information of motion without violating the assumption for DXT-ME. The other advantage of edge extraction is that any change in the illumination condition does not alter the edge information and in turn makes no false motion estimates by the DXT-ME algorithm. Since we only intend to extract the main features of moving objects while keeping the overall complexity low, we employ a very simple edge detection by convolving horizontal and vertical Sobel operators of size 3 × 3 with the image to obtain horizontal and vertical gradients respectively and then combine both gradients by taking the square root of the sum of the squares of both gradients [40]. Edge detection provides us the features of moving objects but also the features of the background (stationary objects), which is undesirable. However, if the features of the background have smaller energy than those of moving objects within every block containing moving objects, then the background features will not affect the performance of DXT-ME. The computational complexity of this preprocessing step is only $O(M^2)$ for a search range $M \times M$ and thus the overall computational complexity is still $O(M^2)$.

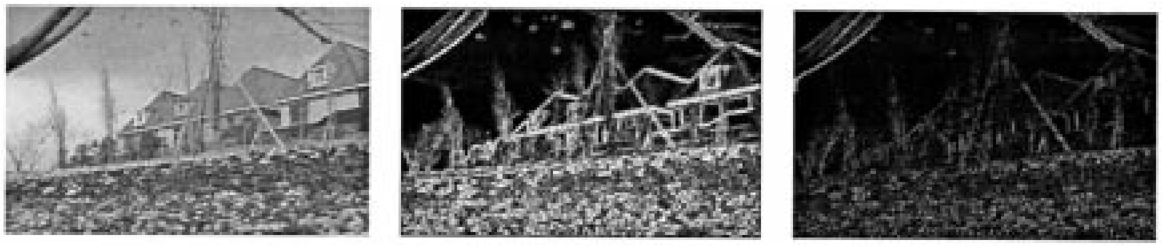
Frame differentiation generates an image of the difference of two consecutive frames. This frame-differentiated image contains no background objects but the difference of moving objects between two frames. The DXT-ME estimator operates directly on this frame-differentiated sequence to predict motion in the original sequence. The estimate will be good if the moving objects are moving constantly in one direction in three consecutive frames. For 30 frames/s, the standard NTSC frame rate, objects can usually be viewed as moving at a constant speed in three consecutive frames. However, for ten frames/s, as commonly found in the videophone applications, the motion may appear jerky and, therefore, may degrade the performance of frame differentiation. Obviously, this step also has only $O(M^2)$ computational complexity.

Preferably, a simple decision rule similar to the one used in the MPEG-1 standard [36] , as depicted in Fig. 9(b), is used to choose among the DXT-ME estimate and no motion. This simplified extended DXT-ME algorithm works very well when combined with the adaptive overlapping approach.

B. Adaptive Overlapping Approach

As the restriction of DXT-ME, the search area must be limited to the size of a candidate block. On the contrary, the block-matching approaches require a larger search area than the candidate block and a larger search area leads to more information available for the motion estimation algorithms. This difference makes the comparison of two different types of methods unfair. For fair comparison with BKM-ME, we adopt the adaptive overlapping approach to enlarge adaptively the block area depending on where the block is located in the whole image, and thus diminish the boundary effect as discussed in Section I.

In Section III, we mention that peaks of DSC and DCS are searched over a fixed index range of interest $\Phi = \{0, \cdots, N/2\}^2$. However, if we follow the partitioning approach used in BKM-ME, then we may dynamically adjust Φ. At first, partition the whole current frame into $bs \times bs$ nonoverlapping reference blocks shown as the shaded area in Fig. 10(a). Each reference block is associated with a larger search area (of size sa) in the previous frame (the dotted region in the same figure) in the same way as for BKM-ME. From the position of a reference block and its associated search area, a search range $\mathcal{D} = \{(u,v)\colon -u_1 \leq u \leq u_2,\ -v_1 \leq v \leq v_2\}$ can then be determined as in Fig. 10(b). Differing from BKM-ME, DXT-ME requires that the reference block size and the search area size must be equal. Thus, instead of using the reference block, we use the block of the same size and position in the current frame as the search area of the previous frame. The peak values of DSC and DCS are searched in a



(a) (b) (c)

Fig. 11. Frame 57 in the flower garden (FG) sequence. (a) Original. (b) Edge extracted. (c) Frame differentiated.

zigzag way as described in Section III over this index range $\Phi = \{0, \cdots, \max(u_2, u_1 - 1)\} \times \{0, \cdots, \max(v_2, v_1 - 1)\}$. In addition to the requirement that the new peak value must be larger than the current peak value by a preset threshold, it is necessary to examine if the motion estimate determined by the new peak index lies in the search region $\mathcal{D}$. Since search areas overlap on one another, the SE-DXT-ME architecture utilizing this approach is called *overlapping SE-DXT-ME*. Even though the block size required by the Overlapping SE-DXT-ME algorithm is larger than the block size for one DCT block, it is still possible to estimate motion completely in the DCT domain without going back to the spatial domain by concatenating neighboring DCT blocks directly in the DCT domain [41].

C. Simulation Results

A number of video sequences with different characteristics are used in our simulations to compare the performance of the DXT-ME algorithm with the full search BKM-ME (or BKM for the sake of brevity) as well as three commonly used fast search block-matching approaches such as the logarithmic search method (LOG), the three step search method (TSS), and the subsampled search approach (SUB) [14]. The performance of different schemes is evaluated and compared in terms of mean squared error per pel (MSE) and bits per sample (BPS) where $\mathrm{MSE} = (\Sigma_{m,n} [\hat{x}(m,n) - x(m,n)]^2/N^2)$ and BPS is the ratio of the total number of bits required for each motion compensated residual frame in JPEG format (BPS) converted by the image format conversion program, ALCHEMY, with quality $=$ 32 to the number of pixels. As widely used in the literature of video coding, all the block-matching methods adopt the conventional MAD optimization criterion

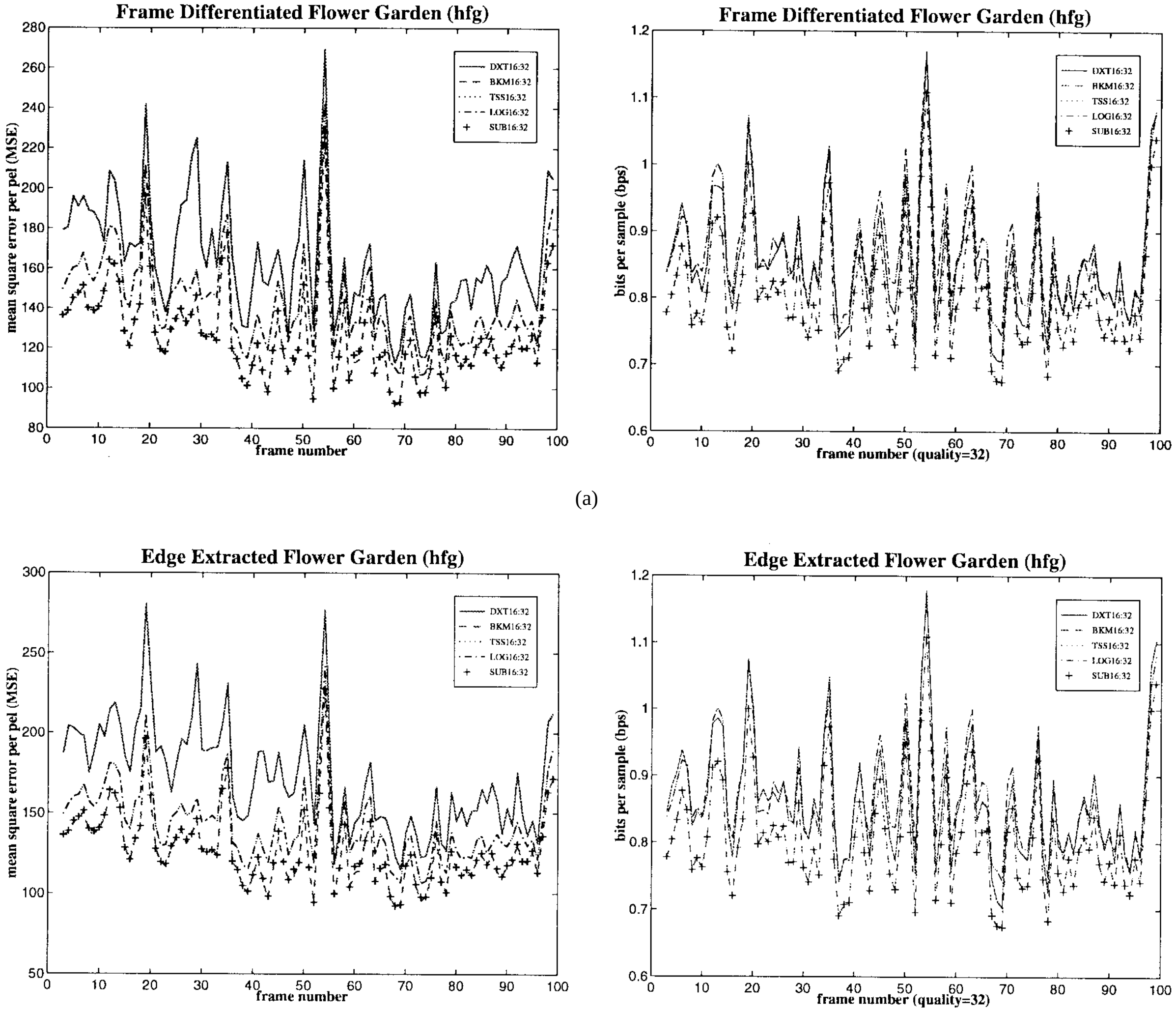
$$\hat{d} = (\hat{u}, \hat{v}) = \arg \min_{(u,v) \in \mathrm{S}} \cdot \frac{\displaystyle\sum_{m,n} |x_2(m,n) - x_1(m-u, n-v)|}{N^2}$$

where S denotes the set of allowable displacements depending on which block-matching approach is in use.

The first sequence is the flower garden (FG) sequence where the camera is moving before a big tree and a flower garden in front of a house as shown in Fig. 11(a). Each frame has 352 × 224 pixels. Simple preprocessing is applied to this sequence: edge extraction or frame differentiation as depicted in Fig. 11(b) and (c), respectively. Since macroblocks, each consisting of 16 × 16 luminance blocks and two 8 × 8 chrominance blocks, are considered to be the basic unit for motion estimation/compensation in MPEG standards [36], the following simulation setting is adopted for simulations on the FG sequence and all other subsequent sequences: 16 × 16 blocks on 32 × 32 search areas. Furthermore, the overlapping SE-DXT-ME algorithm is used for fair comparison with block-matching approaches which require a larger search area.

As can be seen in Fig. 11(b), the edge extracted frames contain significant features of moving objects in the original frames so that DXT-ME can estimate the movement of the objects based upon the information provided by the edge extracted frames. Because the camera is moving at a constant speed in one direction, the moving objects occupy almost the whole scene. Therefore, the background features do not interfere with the operation of DXT-ME much but still affect the overall performance of DXT-ME as compared to the frame-differentiated preprocessing approach. The frame-differentiated images of the FG sequence, one of which is shown in Fig. 11(c), have the residual energy strong enough for DXT-ME to estimate the motion directly on this frame-differentiated sequence due to the constant movement of the camera.

The performances for different motion estimation schemes are plotted in Fig. 12 and summarized in Table III, where the MSE and BPS values of different motion estimation approaches are averaged over the whole sequence from frames 3 to 99 for easy comparison. It should be noted that the MSE difference in Table III is the difference of the MSE value of the corresponding motion estimation scheme from the MSE value of the full search block-matching approach (BKM) and the MSE ratio is the ratio of the MSE difference to the MSE of BKM. As indicated in the performance summary table, the frame differentiated DXT-ME algorithm is 28.9% worse in terms of MSE than the full search block-matching approach while the edge extracted DXT-ME algorithm is 36.0% worse. Surprisingly, even though the fast search block-matching algorithms (only 12.6% worse than BKM), TSS/LOG, have smaller MSE values than the DXT-ME algorithm, TSS/LOG have slightly larger BPS values than the DXT-ME algorithm, as can clearly be seen in Table III and Fig. 12. In other words, the motion-compensated residual frames generated by TSS/LOG require slightly more bits than the DXT-ME algorithm to transmit/store after compression even though the MSE ratios of TSS/LOG are smaller than those of DXT-ME results in this FG sequence.



(a)

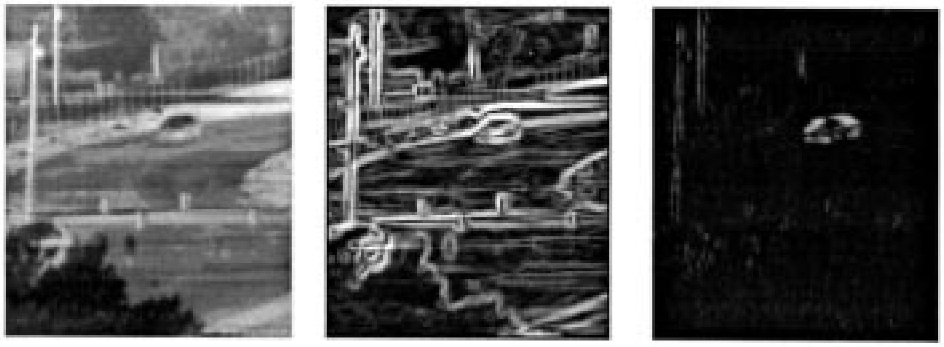
(b)

Fig. 12. Comparison of overlapping SE-DXT-ME with block-matching approaches on FG. (a) Preprocessed with frame differentiation. (b) Preprocessed with edge extraction.

TABLE III
PERFORMANCE SUMMARY OF THE OVERLAPPING SE-DXT-ME ALGORITHM WITH EITHER FRAME DIFFERENTIATION OR EDGE EXTRACTION AS PREPROCESSING AGAINST FULL SEARCH AND FAST SEARCH BLOCK-MATCHING APPROACHES (BKM, TSS, LOG, SUB) OVER THE SEQUENCE FLOWER GARDEN. MSE DIFFERENCE IS THE DIFFERENCE FROM THE MSE VALUE OF FULL SEARCH BLOCK-MATCHING METHOD (BKM) AND MSE RATIO IS THE RATIO OF MSE DIFFERENCE TO THE MSE OF BKM

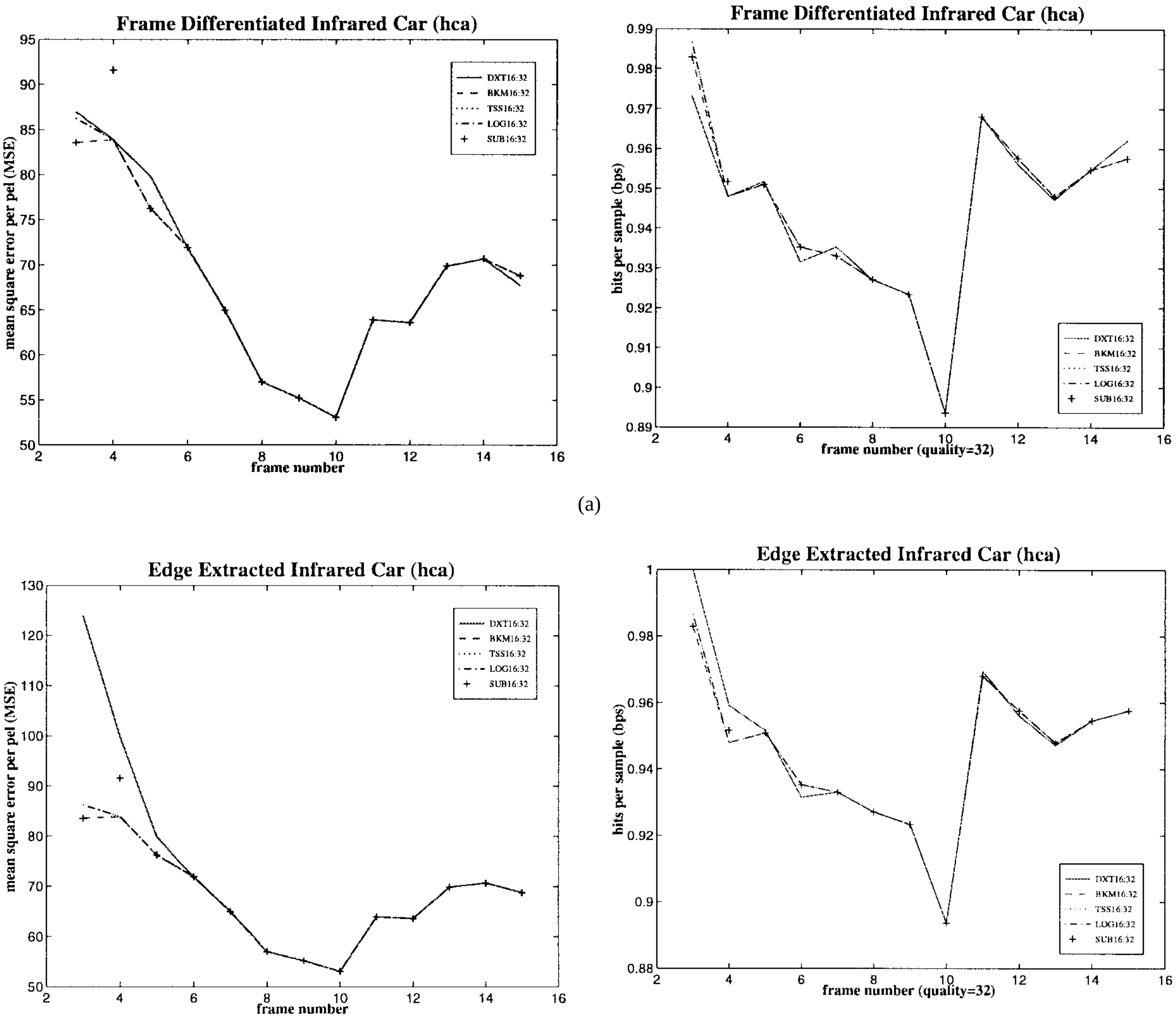
Approach	MSE	MSE difference	MSE ratio	BPF	BPS	BPS ratio
BKM	127.021	0.000	0%	63726	0.808	0%
Frame Differentiated DXT-ME	163.712	36.691	28.9%	67557	0.857	6.0%
Edge Extracted DXT-ME	172.686	45.665	36.0%	68091	0.864	6.8%
TSS	143.046	16.025	12.6%	68740	0.872	7.9%
LOG	143.048	16.026	12.6%	68739	0.872	7.9%
SUB	127.913	0.892	0.7%	63767	0.809	1%

Another simulation is done on the infrared car (CAR) sequence, which has the frame size 96 × 112 and one major moving object, the car moving along the curved road toward the camera fixed on the ground. After preprocessed by edge extraction as shown in Fig. 13(b), the features of both the car and the background are captured in the edge extracted frames. For the first few frames, the features of the roadside behind the car mix with the features of the car moving along the roadside. This mixture is not desirable and hampers the estimation of the DXT-ME algorithm as revealed by the performance plot in Fig. 14 and the performance summary in Table IV. As to the frame differentiated images as shown in Fig. 13(c), the residual energy of the moving car is completely separated from the rest of the scene in most of the preprocessed frames and, therefore, lower MSE values are obtained with this preprocessing function than with edge extraction. In Table IV,



(a) (b) (c)

Fig. 13. Infrared car (CAR) sequence. (a) Original. (b) Edge extracted. (c) Frame differentiated.



(a)

(b)

Fig. 14. Comparison of Overlapping SE-DXT-ME with block-matching approaches on CAR. (a) Preprocessed with frame differentiation. (b) Preprocessed with edge extraction.

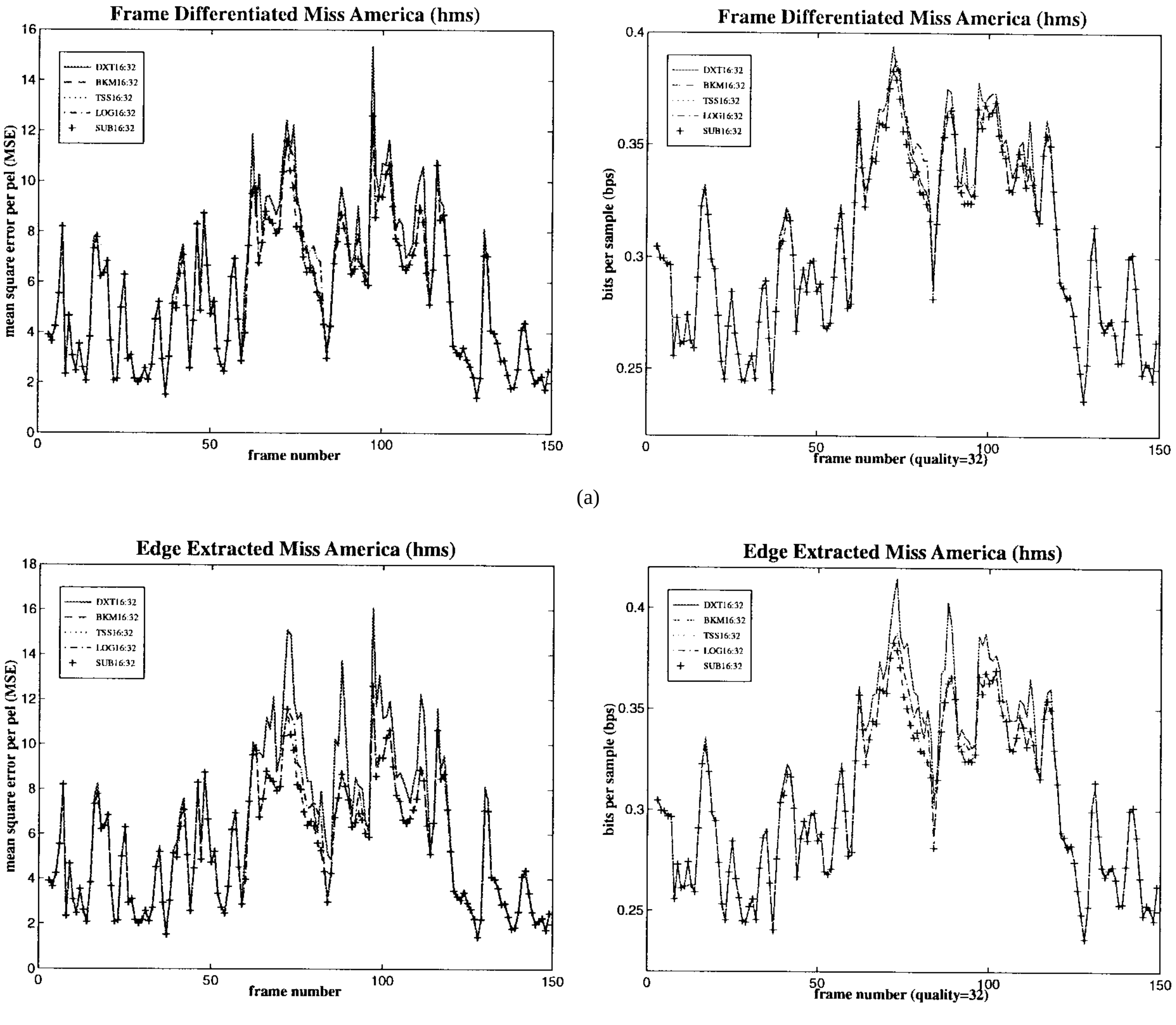
the frame differentiated DXT-ME algorithm is only 0.7% worse than the full search block-matching approach compared to 0.9% for the subsampled approach (SUB) and 0.3% for both LOG and TSS, while the edge extracted DXT-ME has a MSE ratio 6.8%. However, if we compare the BPS values, we find that the frame differentiated DXT-ME requires fewer bits on average for the JPEG compressed residual frames than the full search approach (BKM).

Simulation is also performed on the Miss America sequence in QCIF format, of which each frame has 176 × 144 pixels. This sequence not only has translational motion of the head and shoulders but also the mouth and eyes open and close. This

TABLE IV

PERFORMANCE SUMMARY OF THE OVERLAPPING SE-DXT-ME ALGORITHM WITH EITHER FRAME DIFFERENTIATION OR EDGE EXTRACTION AS PREPROCESSING AGAINST FULL SEARCH AND FAST SEARCH BLOCK-MATCHING APPROACHES (BKM, TSS, LOG, SUB) OVER THE SEQUENCE INFRARED CAR

Approach	MSE	MSE difference	MSE ratio	BPF	BPS	BPS ratio
BKM	67.902	0.000	0%	10156	0.945	0%
Frame Differentiated DXT-ME	68.355	0.453	0.7%	10150	0.944	-0.1%
Edge Extracted DXT-ME	72.518	4.615	6.8%	10177	0.946	0.2%
TSS	68.108	0.206	0.3%	10159	0.945	0.0%
LOG	68.108	0.206	0.3%	10159	0.945	0.0%
SUB	68.493	0.591	0.9%	10159	0.945	0.0%



(a)

(b)

Fig. 15. Comparison of overlapping SE-DXT-ME with block-matching approaches on Miss America in QCIF format.

makes the task of motion estimation difficult for this sequence but the DXT-ME algorithm can still perform reasonably well compared to the block-matching methods, as can be found in Fig. 15. The performance of all the algorithms is summarized in Table V, where the MSE and BPS values are averaged over the whole sequence from frames 3–149. As clearly shown in Table V, the frame differentiated DXT-ME is only 6.9% worse than BKM as compared to 2.1% worse for both LOG and TSS and 0.3% worse for SUB. Furthermore, the BPS achieved by the frame differentiated DXT-ME is 0.307, only 0.9% larger than BKM. However, the edge extracted DXT-ME performs somewhat worse than the frame-differentiated DXT-ME and achieves 2% more of MSE than BKM.

From all the above simulations, it seems that frame differentiation is a better choice for preprocessing than edge extraction due to its capability of removing background features, which in some cases affect adversely the performance of the DXT-ME algorithm.

D. Rough Count of Computations

In the previous section, we choose the asymptotic complexity for comparison because calculation of the actual number

TABLE V

PERFORMANCE SUMMARY OF THE OVERLAPPING SE-DXT-ME ALGORITHM WITH EITHER FRAME DIFFERNTIATION OR EDGE EXTRACTION AS PREPROCESSING AGAINST FULL SEARCH AND FAST SEARCH BLOCK-MATCHING APPROACHES (BKM, TSS, LOG, SUB) OVER THE SEQUENCE MISS AMERICA IN QCIF FORMAT

Approach	MSE	MSE difference	MSE ratio	BPF	BPS	BPS ratio
BKM	5.448	0.000	0%	7714	0.304	0%
Frame Differentiated DXT-ME	5.823	0.374	6.9%	7786	0.307	0.9%
Edge Extracted DXT-ME	6.229	0.781	14.3%	7865	0.310	2.0%
TSS	5.561	0.112	2.1%	7749	0.306	0.5%
LOG	5.561	0.113	2.1%	7749	0.306	0.5%
SUB	5.466	0.017	0.3%	7716	0.304	0.0%

of computations requires the knowledge of specific implementations, which is totally different from the simple block-matching methods, whose implementations are simple and computations can be counted without the knowledge of the actual architectures. However, in application of the DXT-ME algorithm to video coding in which block-matching methods, either full search or fast search, are commonly employed, we try to make a rough count of computations required by the algorithm based on the straight forward software implementation.

In DCT-based motion-compensated video coding, DCT, IDCT and peak searching are required, and therefore we will count only the number of operations required in the pseudophase computation. At each pixel position, we need to solve a 4 × 4 linear equation by means of the Gauss elimination method with four divisions, 40 multiplications, and 30 additions/subtractions. Therefore, the total number of operations is 18 944 for a 16 × 16 block and 75776 for a corresponding overlapped block (32 × 32), while the BKM-ME approach requires 130 816 additions/subtractions for block size 16 × 16 and search area 32 × 32. Still, the number of operations required by the DXT-ME algorithm is smaller than BKM-ME. Further reduction of computations can be achieved by exploiting various properties in the algorithm. For example, if the denominator is found to be ill-conditioned, it is possible to skip any further computation and set the pseudophase at that index position as zero. In this way, the required number of operations is reduced. Of course, the exact number of required operations must be counted based on the actual implementation or architecture, which is beyond the topic in this paper.

V. CONCLUSION

In this paper, we present new DCT pseudophase techniques utilizing the concept of pseudophase shifts hidden in the DCT coefficients of shifted signals and the sinusoidal orthogonal principles for estimation of shift/delay completely in the DCT domain. In extension to the 2-D case, the DCT pseudophase techniques result in the DXT-ME algorithm, a DCT-based motion estimation algorithm. We show that this DXT-ME algorithm exhibits good estimates even in a noisy situation. Due to its capability of motion estimation in the DCT domain, its application to video coding realizes the fully DCT-based motion-compensated video coder structure which contains, in the performance-critical feedback loop of the coder, only one major component, the transform-domain motion estimation unit instead of three major components as in the conventional hybrid DCT motion-compensated video coder design, and thus achieves higher throughput and lower system complexity. In addition to this advantage, the DXT-ME algorithm has low computational complexity: $O(M^2)$ as compared to $O(N^2 \cdot M^2)$ for the full search block-matching approach (BKM-ME) or 75 776 operations versus 130 816 operations for BKM-ME depending on the actual implementation. Even though the DXT-ME algorithm is not in the category of the fast search block-matching schemes, we compare its performance with BKM-ME and some fast search approaches such as three step search (TSS), logarithmic search (LOG), and subsampled search (SUB), and find that for the FG and CAR sequences, the DXT-ME algorithm achieves fewer BPS of the motion-compensated residual images (DFD) than all other fast search approaches while for other sequences, the DXT-ME algorithm shows higher BPS than other fast block search approaches. Finally, its DCT-based nature enables us to incorporate its implementation with the DCT codecs design to gain further savings in complexity and this DXT-ME algorithm has inherently highly parallel operations in computing the pseudophases very suitable for VLSI implementation.

REFERENCES

[1] J. K. Aggarwal and N. Nandhakumar, “On the computation of motion from sequences of images—A review,” *Proc. IEEE*, vol. 76, pp. 917–935, Aug. 1988.

[2] H. G. Musmann, P. Pirsch, and H.-J. Grallert, “Advances in picture coding,” *Proc. IEEE*, vol. 73, pp. 523–548, Apr. 1993.

[3] K. M. Yang, M. T. Sun, and L. Wu, “A family of VLSi designs for the motion compensation block-matching algorithm,” *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1317–1325, Oct. 1989.

[4] T. Komarek and P. Pirch, “Array architectures for block-matching algorithms,” *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1301–1308, Oct. 1989.

[5] L. D. Vos and M. Stegherr, “Parametrizable VLSI architectures for the full-search block-matching algorithm,” *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1309–1316, Oct. 1989.

[6] R. C. Kim and S. U. Lee, “A VLSI architecture for a pel recursive motion estimation algorithm,” *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1291–1300, Oct. 1989.

[7] *IEEE Trans. Signal Processing, Spec. Issue Time Delay Estimat.*, vol. ASSP-29, 1981.

[8] G. Jacovitti and G. Scarano, “Discrete-time techniques for time-delay estimation,” *IEEE Trans. Signal Processing*, vol. 41, pp. 525–533, 1993.

[9] J. P. Fillard, J. M. Lussert, M. Castagne, and H. M’timet, “Fourier phase shift location estimation of unfocused optical point spread functions,” *Signal Process.: Image Commun.*, vol. 6, pp. 281–287, Aug. 1994.

[10] J. R. Jain and A. K. Jain, “Displacement measurement and its application in interframe image coding,” *IEEE Trans. Commun.*, vol. COMM-29, pp. 1799–1806, Dec. 1981.

[11] T. Koga *et al.*, “Motion-compensated interframe coding for video conferencing,” in *Proc. Nat. Telecommunications Conf.*, New Orleans, LA, Dec. 1981, pp. G5.3.1–G.5.3.5.

[12] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," *IEEE Trans. Commun.*, vol. COMM-33, no. 8, pp. 888–896, Aug. 1985.

[13] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol. 38, pp. 950–953, July 1990.

[14] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 148–157, Apr. 1993.

[15] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 438–442, Aug. 1994.

[16] R. W. Young and N. G. Kingsbury, "Frequency-domain motion estimation using a complex lapped transform," *IEEE Trans. Image Processing*, vol. 2, pp. 2–17, Jan. 1993.

[17] A. N. Netravali and J. D. Robbins, "Motion compensated television coding—Part 1," *Bell Syst. Tech. J.*, vol. 58, pp. 631–670, Mar. 1979.

[18] J. D. Robbins and A. N. Netravali, "Recursive motion compensation: A review," in *Image Sequence Processing and Dynamic Scene Analysis*, T. S. Huang, Ed. Berlin, Germany: Springer-Verlag, 1983, pp. 76–103.

[19] A. Singh, *Optic Flow Computation—A Unified Perspective*. New York: IEEE Comput. Soc. Press, 1991.

[20] C. D. Kughlin and D. C. Hines, "The phase correlation image alignment method," in *Proc. 1975 IEEE Int. Conf. Systems, Man, and Cybernetics*, Sept. 1975, pp. 163–165.

[21] G. A. Thomas, "Television motion measurement for DATV and other applications," Tech. Rep. 11, BBC Res. Dept., 1987.

[22] B. Girod, "Motion-compensating prediction with fractional-pel accuracy," *IEEE Trans. Commun.*, vol. 41, p. 604, Apr. 1993.

[23] B. Porat and B. Friedlander, "A frequency domain algorithm for multiframe detection and estimation of dim targets," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 398–401, Apr. 1990.

[24] A. Kojima, N. Sakurai, and J. Kishigami, "Motion detection using 3D-FFT spectrum," in *Proc. ICASSP'39*, Apr. 1993, vol. V, pp. V213–V216.

[25] D. Heeger, "A model for extraction of image flow," in *Proc. 1st Int. Conf. Comput. Vis.*, London, U.K., 1987, pp. 181–190.

[26] U. V. Koc and K. J. R. Liu, "Discrete-cosine/sine-transform based motion estimation," in *Proc. IEEE Int. Conf. Image Processing (ICIP'94)*, Austin, TX, Nov. 1994, vol. 3, pp. 771–775.

[27] ——, "Adaptive overlapping approach for DCT-based motion estimation," in *Proc. IEEE Int. Conf. Image Processing*, Washington, D.C., vol. I, pp. 223–226.

[28] U. V. Koc, "Low complexity and high throughput fully DCT-based motion compensated video coders," Ph.D. dissertation, Univ. Maryland, College Park, July 1996.

[29] S.-F. Chang and D. G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video," *IEEE J. Select. Areas Commun.*, vol. 13, p. 1, Jan. 1995.

[30] N. Merhav and V. Bhaskaran, "A fast algorithm for DCT-domain inverse motion compensation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 1996, vol. IV, pp. 2309–2312.

[31] P. Yip and K. R. Rao, "On the shift property of DCT's and DST's," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 404–406, Mar. 1987.

[32] C. T. Chiu and K. J. R. Liu, "Real-time parallel and fully pipelined two-dimensional DCT lattice structures with applications to HDTV systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, pp. 25–37, Mar. 1992.

[33] K. J. R. Liu and C. T. Chiu, "Unified parallel lattice structures for time-recursive discrete cosine/sine/Hartley transforms," *IEEE Trans. Signal Processing*, vol. 41, pp. 1357–1377, Mar. 1993.

[34] K. J. R. Liu, C. T. Chiu, R. K. Kologotla, and J. F. JaJa, "Optimal unified architectures for the real-time computation of time-recursive discrete sinusoidal transforms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 168–180, Apr. 1994.

[35] CCITT Recommendation H.261, *Video Codec for Audiovisual Services at $p \times 64$ kbit/s*, CCITT, Aug. 1990.

[36] CCITT Recommendation MPEG-1, *Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s*, ISO/IEC 11172, Geneva, Switzerland, 1993.

[37] J. S. McVeigh and S.-W. Wu, "Comparative study of partial closed-loop versus open-loop motion estimation for coding of HDTV," in *Proc. IEEE Workshop on Visual Signal Processing and Communications*, New Brunswick, NJ, Sept. 1994, pp. 63–68.

[38] H. Li, A. Lundmark, and R. Forchheimer, "Image sequence coding at very low bitrates: A review," *IEEE Trans. Image Processing*, vol. 3, pp. 589–608, Sept. 1994.

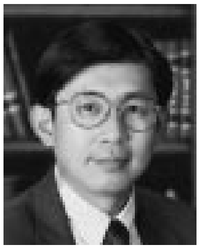
[39] A. Zakhor and F. Lari, "Edge-based 3-d camera motion estimation with application to video coding," *IEEE Trans. Image Processing*, vol. 2, pp. 481–498, Oct. 1993.

[40] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[41] W. D. Kou and T. Fjallbrant, "A direct computation of DCT coefficients for a signal block taken from 2 adjacent blocks," *IEEE Trans. Signal Processing*, vol. 39, pp. 1692–1695, July 1991.

Ut-Va Koc (S'91–M'96) received the B.S. degree in electronics engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, R.O.C., in 1989, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park (UMCP), in 1992 and 1996, respectively.

From 1989 to 1990, he was a Teaching Assistant at NCTU. From 1991 to 1992, he was with the Plasma Research Center, UMCP. From 1992 through 1996, he was a Research Assistant at the Institute for Systems Research and the Lab Manager of Digital Signal Processing Laboratory. He is currently with Bell Laboratories, Lucent Technologies, Murray Hill, NJ, as Member of Technical Staff. His research interests include video compression, source/channel coding, communications, digital signal processing algorithms/architectures, transceivers, adaptive digital filters, and VLSI design.



K. J. Ray Liu (S'86–M'90–SM'93) received the B.S. degree from National Taiwan University, Taipei, Taiwan, R.O.C., in 1983, and the Ph.D. degree from the University of California, Los Angeles, in 1990, both in electrical engineering.

Since 1990, he has been with the Electrical Engineering Department and Institute for Systems Research, University of Maryland, College Park, where he is an Associate Professor. During his sabbatical leave in 1996–1997, he was Visiting Associate Professor at Stanford University, Stanford, CA. His research interests span various aspects of signal/image processing and communications. He has published over 130 papers, of which over 50 are in archival journals, books, and book chapters. His research web page is at http://dspserv.eng.umd.edu.

Dr. Liu has been an Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING. He is currently a Guest Editor of special issues on Multimedia Signal Processing and Technology of PROCEEDINGS OF THE IEEE. He is a Guest Editor of the special issue on Signal Processing for Wireless Communications of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, an Editor of *Journal of VLSI Signal Processing*, and a founding member of Multimedia Signal Processing Technical Committee of the IEEE Signal Processing Society. He is the Co-editor of *High Performance VLSI Signal Processing: Volume I: System Design and Methodology; Vol. II: Algorithms, Architectures, and Applications* (IEEE Press and SPIE). He has received numerous awards, including the 1994 National Science Foundation Young Investigator Award; the IEEE Signal Processing Society's 1993 Senior Award (Best Paper Award); and the George Corcoran Award in 1994 for outstanding contributions to electrical engineering education, and the 1995–96 Outstanding Systems Engineering Faculty Award in recognition of outstanding contributions in interdisciplinary research, both from the University of Maryland.