

# Interpolation-Free Subpixel Motion Estimation Techniques in DCT Domain

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

**Abstract**—Currently existing subpixel motion estimation algorithms require interpolation of interpixel values which undesirably increases the overall complexity and data flow and deteriorates estimation accuracy. In this paper, we develop discrete cosine transform (DCT)-based techniques to estimate subpel motion at different desired subpel levels of accuracy in the DCT domain without interpolation. We show that subpixel motion information is preserved in the DCT of a shifted signal under some condition in the form of pseudophases, and we establish subpel sinusoidal orthogonal principles to extract this information. The proposed subpixel techniques are flexible and scalable in terms of estimation accuracy with very low computational complexity  $O(N^2)$  compared to  $O(N^4)$  for the full-search block-matching approach and its subpixel versions. Above all, motion estimation in the DCT domain instead of the spatial domain simplifies the conventional hybrid DCT-based video coder, especially the heavily loaded feedback loop in the conventional design, resulting in a fully DCT-based high-throughput video codec. In addition, the computation of pseudophases is local, and thus a highly parallel architecture is feasible for the DCT-based algorithms. Finally, simulation on video sequences of different characteristics shows comparable performance of the proposed algorithms to block-matching approaches.

**Index Terms**—Motion estimation, shift measurement, subpixel accuracy, video coding, video compression.

## I. INTRODUCTION

ACCURATE estimation of displacement or location of a signal or image is important in many applications of signal and image processing such as time delay estimation [22], target tracking [37], noncontact measurement [2], [41], remote sensing [4], [11], computer vision [1], image registration [8], [39], and so on. In video coding, motion estimation is proven to be very useful for the reduction of temporal redundancy. Therefore, a number of motion estimation algorithms have been devised solely for video coding [10], [31], and numerous VLSI architectures have been designed for practical video applications [35]. To further improve the compression rate, motion estimation with subpixel accuracy is essential because movements in a video sequence are not necessarily multiples of the sampling grid distance in the rectangular sampling grid of a camera. It is shown that significant improvement of coding

gain can be obtained with motion estimation of half-pixel or finer accuracy [16]. Further investigation reveals that the temporal prediction error variance is generally decreased by subpixel motion compensation, but beyond a certain “critical accuracy,” the possibility of further improving prediction by more accurate motion compensation is small [13]. As suggested in [12], and [16], motion compensation with 1/4-pel accuracy is sufficiently accurate for broadcast TV signals, but for videophone signals, half-pel accuracy is good enough. As a result, motion compensation with half-pel accuracy is recommended in MPEG standards [29], [30]. Implementations of half-pel motion estimation have started to be realized [3], [6], [40].

Many subpixel motion estimation schemes have been proposed over the years [1], [10], [31]. The most commonly used spatial-domain fractional-pel motion estimation algorithms such as the block-matching approach [9], [12], [28], and the pel-recursive approach [32], [33] require interpolation of images through bilinear, Lagrange, or other interpolation methods [36]. However, interpolation not only increases the complexity and data flow of a coder, but also may adversely affect the accuracy of motion estimates from the interpolated images [12]. It is more desirable that subpixel accuracy of motion estimates can be obtained without interpolating the images. In the category of frequency-domain methods, the phase correlation technique [38], [42], [23] is reported to provide accurate estimates without interpixel interpolation, but is based on the fast Fourier transform (FFT), which is incompatible with discrete cosine transform (DCT)-based video coding standards and requires a large search window at a high computational cost. Other FFT-based approaches such as in [17], [21] also have similar drawbacks.

Due to the fact that the motion-compensated DCT-based hybrid approach is the backbone of several international video coding standards such as CCITT H.261 [14], MPEG1 [29], MPEG2 [30], and the emerging HDTV [5] and H.263 [15] standards. It is more desirable to estimate motion with fractional-pel accuracy *without* any interpixel interpolation at a low computational cost in the DCT domain so that seamless integration of the motion compensation unit with the spatial compression unit is possible. More specifically, a conventional standard-compliant video coder is usually implemented as a hybrid DCT-based structure in Fig. 1(a), which achieves spatial compression through the DCT and temporal compression through motion compensation traditionally accomplished in the spatial domain. In this hybrid structure, the feedback loop contains three major components:

Manuscript received October 28, 1996; revised March 9, 1998. This work was supported in part by ONR Grant N00014-93-1-0566, NSF Grant MIP9457397, and MIPS/MicroStar. This paper was recommended by Associate Editor D. Anastassiou.

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 the Institute for Systems Research, University of Maryland, College Park, MD 20742 USA (e-mail: kjrlu@eng.umd.edu).

Publisher Item Identifier S 1051-8215(98)05763-2.

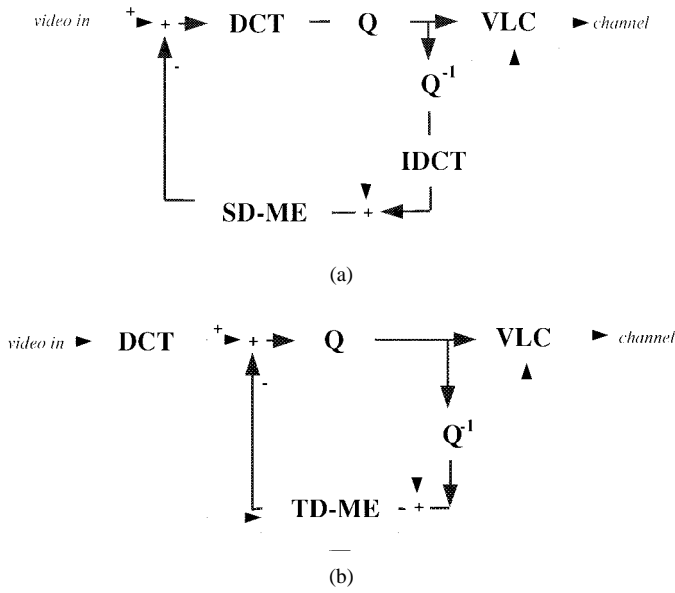


Fig. 1. Encoder structures. (a) Commonly used motion-compensated DCT hybrid coder performs motion estimation in the spatial domain. (b) Fully DCT-based coder estimates motion in the transform domain.

DCT, IDCT (inverse DCT), and SD-ME (spatial domain motion estimation). All incoming raw video data must traverse this heavily loaded feedback loop once in order to be encoded in the output bit stream. In addition to the disadvantage of having more hardware components, the throughput of the whole coder is also limited by the complexity of the loop. However, if motion can be estimated and compensated entirely in the transform domain, then DCT can be moved out of the loop and IDCT can be eliminated, resulting in a fully DCT-based video coder as shown in Fig. 1(b) where the feedback loop has only one major component, transform domain motion estimation (TD-ME) [24], instead of three major components.

Based upon the concept of pseudophases in DCT coefficients and the sinusoidal orthogonal principles, a DCT-based integer-pel motion estimation scheme (DXT-ME) of very low computational complexity ( $O(N^2)$  as opposed to  $O(N^4)$  for the widely used full search block-matching algorithm) was proposed in [19], [20] to realize the fully DCT-based video coder design, as depicted in Fig. 2 and summarized in Table I. In this paper, we further explore this DCT-based concept at the subpixel level, and show that if the spatial sampling of images satisfies the Nyquist criterion, the subpixel motion information is preserved in the pseudophases of DCT coefficients of moving images. Furthermore, it can be shown that with appropriate modification, the sinusoidal orthogonal principles can still be applicable, except that an impulse function is replaced by a sinc function whose peak position reveals subpixel displacement. Therefore, exact subpixel motion displacement can be obtained without the use of interpolation. From these observations, we can develop a set of subpixel DCT-based motion estimation algorithms that are fully compatible with the integer-pel motion estimator for low-complexity and high-throughput video applications.

In this paper, we discuss the pseudophases carrying subpixel motion information in Section II, and the subpel sinusoidal

orthogonal principles in Section III for objects moving out of synchronization with the sampling grid. In Section IV, we propose the DCT-based half-pel (HDXT-ME) and quarter-pel (QDXT-ME and Q4DXT-ME) motion estimation algorithms whose simulation results on actual video sequences of different characteristics are presented in Section V in comparison with the popular block-matching approaches. Finally, we conclude the major contributions of this paper in Section VI.

## II. PSEUDOPHASES AT SUBPIXEL LEVEL

### A. One-Dimensional Signal Model

Without loss of generality, let us consider the one-dimensional model in which a continuous signal  $x_c(t)$  and its shifted version  $x_c(t-d)$  are sampled at a sampling frequency  $1/T$  to generate two sample sequences  $\{x_1(n) = x_c(nT)\}$  and  $\{x_2(n) = x_c(nT-d)\}$ , respectively. Let us define the DCT and DST coefficients as

$$X_i^C(k) \triangleq \text{DCT}\{x_i\} = \frac{2C(k)}{N} \sum_{n=0}^{N-1} x_i(n) \cos \frac{k\pi}{N} \left(n + \frac{1}{2}\right) \quad (1)$$

$$X_i^S(k) \triangleq \text{DST}\{x_i\} = \frac{2C(k)}{N} \sum_{n=0}^{N-1} x_i(n) \sin \frac{k\pi}{N} \left(n + \frac{1}{2}\right) \quad (2)$$

where

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

for  $i = 1$  or  $2$ . By using the sinusoidal relationship

$$\begin{aligned} \cos \frac{k\pi}{N} \left(n + \frac{1}{2}\right) &= \frac{1}{2} \left[ e^{j(k\pi/N)(n+(1/2))} + e^{-j(k\pi/N)(n+(1/2))} \right]; \\ \sin \frac{k\pi}{N} \left(n + \frac{1}{2}\right) &= \frac{1}{2j} \left[ e^{j(k\pi/N)(n+(1/2))} - e^{-j(k\pi/N)(n+(1/2))} \right] \end{aligned} \quad (3)$$

we can show that the DCT/DST and DFT coefficients are related as follows:

$$X_i^C(k) = \frac{C(k)}{N} \left[ \tilde{X}_i^Z(-k) e^{j(k\pi/2N)} + \tilde{X}_i^Z(k) e^{-j(k\pi/2N)} \right] \quad \text{for } k = 0, \dots, N-1 \quad (4)$$

$$X_i^S(k) = \frac{C(k)}{jN} \left[ \tilde{X}_i^Z(-k) e^{j(k\pi/2N)} - \tilde{X}_i^Z(k) e^{-j(k\pi/2N)} \right] \quad \text{for } k = 1, \dots, N \quad (5)$$

where  $\{\tilde{X}_i^Z(k)\}$  is the DFT of the zero-padded sequence  $\{x_i^Z(n)\}$  defined as

$$x_i^Z(n) = \begin{cases} x_i(n), & \text{for } n = 0, \dots, N-1 \\ 0, & \text{for } n = N, \dots, 2N-1 \end{cases} \quad (6)$$

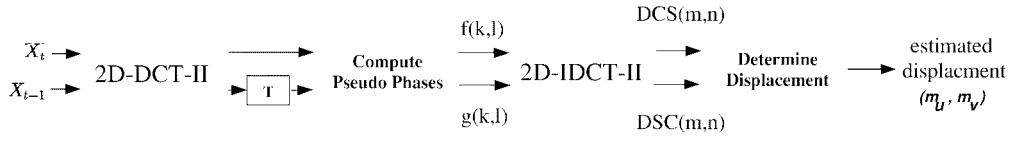


Fig. 2. Block diagram of the DCT-based integer-pel motion estimator (DXT-ME).

TABLE I  
SUMMARY OF DCT-BASED INTEGER-PEL MOTION ESTIMATION SCHEME (DXT-ME)

1. Compute the 2-D DCT coefficients of second kind (2D-DCT-II) of a  $N \times N$  block of pixels at the current frame  $t$ ,  $\{x_t(m, n); m, n \in \{0, \dots, N-1\}\}$ .
2. Convert the stored 2D-DCT-II coefficients of the corresponding  $N \times N$  block of pixels at the previous frame  $t-1$ ,  $\{x_{t-1}(m, n); m, n \in \{0, \dots, N-1\}\}$  to 2D DCT coefficients of first kind (2D-DCT-I) through a simple rotation unit T.
3. Find the pseudo phases  $\{g^{CS}(k, l); k = 0, 1, \dots, N-1; l = 1, 2, \dots, N\}$  and  $\{g^{SC}(k, l); k = 1, 2, \dots, N; l = 0, 1, \dots, N-1\}$ , which are calculated from the DCT coefficients independently at each spectral location  $(k, l)$ .
4. Determine the normalized pseudo phases  $f(k, l)$  and  $g(k, l)$  from  $g^{CS}(k, l)$  and  $g^{SC}(k, l)$  respectively by setting ill-formed  $g^{CS}(k, l)$  and  $g^{SC}(k, l)$  to zero:

$$f(k, l) = \begin{cases} C(k)C(l)g^{CS}(k, l), & \text{for } |g^{CS}(k, l)| \leq 1, \\ 0, & \text{otherwise,} \end{cases}$$

$$g(k, l) = \begin{cases} C(k)C(l)g^{SC}(k, l), & \text{for } |g^{SC}(k, l)| \leq 1, \\ 0, & \text{otherwise,} \end{cases}$$

where

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

5. Obtain the inverse DCT (2D-IDCT-II) of  $f(k, l)$  and  $g(k, l)$  as  $DCS(m, n)$  and  $DSC(m, n)$  for  $m, n \in \{0, \dots, N-1\}$  respectively which basically are composed of impulse functions whose peak positions indicate the shift amount and peak signs reveal the direction of the movement:

$$DCS(m, n) = \frac{4}{N^2} \sum_{k=0}^{N-1} \sum_{l=1}^N C(k)C(l)f(k, l) \cos \frac{k\pi}{N}(m + \frac{1}{2}) \sin \frac{l\pi}{N}(n + \frac{1}{2}),$$

$$DSC(m, n) = \frac{4}{N^2} \sum_{k=1}^N \sum_{l=0}^{N-1} C(k)C(l)g(k, l) \sin \frac{k\pi}{N}(m + \frac{1}{2}) \cos \frac{l\pi}{N}(n + \frac{1}{2}).$$

6. Search for the peaks of  $DCS(m, n)$  and  $DSC(m, n)$  over  $(m, n) \in \{0, \dots, N-1\}^2$  (or range of interest) such that

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

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

7. Estimate the displacement  $\hat{d} = (\hat{m}_u, \hat{m}_v)$  from the signs and positions of the peaks of  $DCS(m, n)$  and  $DSC(m, n)$ :

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

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

so that

$$\tilde{X}_i^Z(k) \triangleq \text{DFT}\{x_i^Z\} = \sum_{n=0}^{N-1} x_i(n) e^{-j(2k\pi n/2N)},$$

for  $k = 0, \dots, 2N-1$ . (7)

From the sampling theorem, we know that the discrete-time Fourier transform (DTFT) of sequences  $x_1(n)$  and  $x_2(n)$  are

related to the Fourier transform (FT) of  $x_c(t)$ ,  $X_c(\Omega)$  in the following way:

$$X_1(\omega) \triangleq \text{DTFT}\{x_1\} = \frac{1}{T} \sum_l X_c\left(\frac{\omega - 2\pi l}{T}\right) \quad (8)$$

$$X_2(\omega) \triangleq \text{DTFT}\{x_2\} = \frac{1}{T} \sum_l X_c\left(\frac{\omega - 2\pi l}{T}\right) e^{-j(\omega - 2\pi l/T)d}. \quad (9)$$

Furthermore, if  $X_c(\Omega)$  is bandlimited in the baseband  $(-\pi/T, \pi/T)$ , then for  $\Omega = (\omega/T) \in (-\pi/T, \pi/T)$ ,

$$X_1(\Omega T) = \frac{1}{T} X_c(\Omega) \quad (10)$$

$$X_2(\Omega T) = \frac{1}{T} X_c(\Omega) e^{-j\Omega d}. \quad (11)$$

Thus, the DFT's of  $x_1(n)$  and  $x_2(n)$  are

$$\begin{aligned} \tilde{X}_1(k) &\triangleq \text{DFT}\{x_1\} = \sum_{n=0}^{N-1} x_1(n) e^{-j(2\pi kn/N)} \\ &= X_1\left(\frac{2\pi k}{N}\right) = \frac{1}{T} X_c\left(\frac{2\pi k}{NT}\right) \end{aligned} \quad (12)$$

$$\begin{aligned} \tilde{X}_2(k) &\triangleq \text{DFT}\{x_2\} = \sum_{n=0}^{N-1} x_2(n) e^{-j(2\pi kn/N)} \\ &= X_2\left(\frac{2\pi k}{N}\right) = \frac{1}{T} X_c\left(\frac{2\pi k}{NT}\right) e^{-j(2\pi kd/NT)} \end{aligned} \quad (13)$$

whereas the DFT's of  $x_1^Z(n)$  and  $x_2^Z(n)$  become

$$\tilde{X}_1^Z(k) = X_1\left(\frac{\pi k}{N}\right) = \frac{1}{T} X_c\left(\frac{\pi k}{NT}\right) \quad (14)$$

$$\tilde{X}_2^Z(k) = X_2\left(\frac{\pi k}{N}\right) = \frac{1}{T} X_c\left(\frac{\pi k}{NT}\right) e^{-j(\pi kd/NT)}. \quad (15)$$

Therefore

$$X_2\left(\frac{\pi k}{N}\right) = X_1\left(\frac{\pi k}{N}\right) e^{-j(\pi kd/NT)}. \quad (16)$$

Substituting (16) back into (4)–(5), we get

$$\begin{aligned} X_2^C(k) &= \frac{C(k)}{N} \left[ \tilde{X}_1^Z(-k) e^{j(k\pi d/NT)} e^{j(k\pi/2N)} \right. \\ &\quad \left. + \tilde{X}_1^Z(k) e^{-j(k\pi d/NT)} e^{-j(k\pi/2N)} \right], \\ &\quad \text{for } k = 0, \dots, N-1, \end{aligned} \quad (17)$$

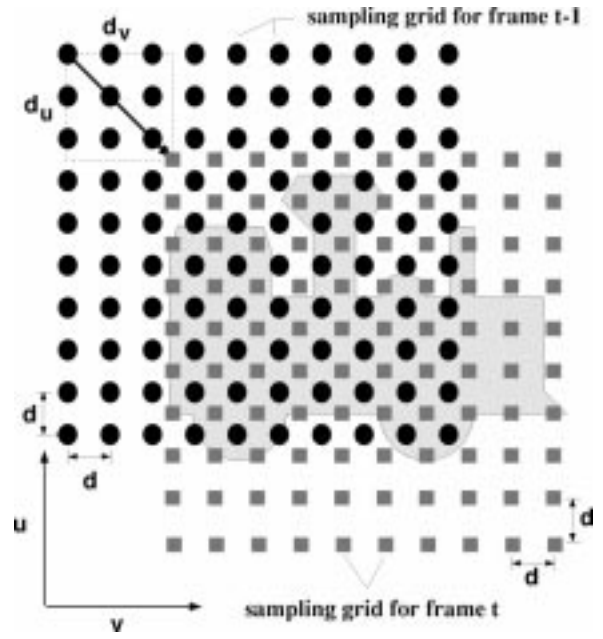
$$\begin{aligned} X_2^S(k) &= \frac{C(k)}{jN} \left[ \tilde{X}_1^Z(-k) e^{j(k\pi d/NT)} e^{j(k\pi/2N)} \right. \\ &\quad \left. - \tilde{X}_1^Z(k) e^{-j(k\pi d/NT)} e^{-j(k\pi/2N)} \right], \\ &\quad \text{for } k = 1, \dots, N. \end{aligned} \quad (18)$$

Using the sinusoidal relationship in (3) to change natural exponents back to cosine/sine, we finally obtain the relationship between  $x_1(n)$  and  $x_2(n)$  in the DCT/DST domain:

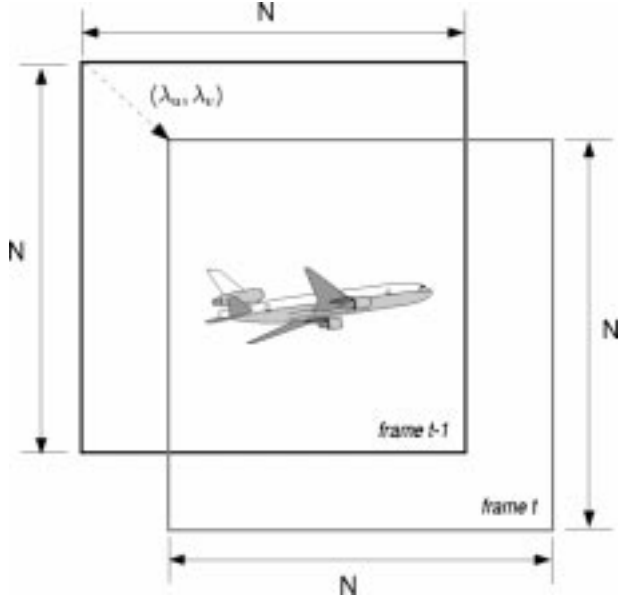
$$\begin{aligned} X_2^C(k) &= \frac{2C(k)}{N} \sum_{n=0}^{N-1} x_1(n) \cos \frac{k\pi}{N} \left( n + \frac{d}{T} + \frac{1}{2} \right), \\ &\quad \text{for } k = 0, \dots, N-1 \end{aligned} \quad (19)$$

$$\begin{aligned} X_2^S(k) &= \frac{2C(k)}{N} \sum_{n=0}^{N-1} x_1(n) \sin \frac{k\pi}{N} \left( n + \frac{d}{T} + \frac{1}{2} \right), \\ &\quad \text{for } k = 1, \dots, N. \end{aligned} \quad (20)$$

We conclude the result in the following theorem.



(a)



(b)

Fig. 3. (a) Black dots and the gray squares symbolize the sampling grids for frames  $I_{t-1}(u, v)$  and  $I_t(u, v)$  at a sampling distance  $d$ , respectively. These two frames are aligned on the common object displaced by  $(d_u, d_v)$  in the continuous coordinate  $(u, v)$ . (b) Two digitized images of consecutive frames  $x_{t-1}(m, n)$  and  $x_t(m, n)$  are aligned on the common object moving  $(\lambda_u, \lambda_v) = (d_u/d, d_v/d)$  pixels southeast.

**Theorem 1:** If a continuous signal  $x_c(t)$  is  $(\pi/T)$ -bandlimited and the sampled sequences of  $x_c(t)$  and  $x_c(t-d)$  are  $\{x_c(nT)\}$  and  $\{x_c(nT-d)\}$ , respectively, then their DCT and DST are related by

$$\text{DCT}\{x_c(nT-d)\} = \text{DCT}_{d/T}\{x_c(nT)\} \quad (21)$$

$$\text{DST}\{x_c(nT-d)\} = \text{DST}_{d/T}\{x_c(nT)\} \quad (22)$$

where

$$\text{DCT}_\alpha\{x\} \triangleq \frac{2C(k)}{N} \sum_{n=0}^{N-1} x(n) \cos \frac{k\pi}{N} \left( n + \alpha + \frac{1}{2} \right) \quad (23)$$

TABLE II  
DETERMINATION OF DIRECTION OF MOVEMENT  
( $\lambda_u, \lambda_v$ ) FROM THE SIGNS OF  $\overline{DSC}$  AND  $\overline{DCS}$

Sign of $\overline{DSC}$ Peak	Sign of $\overline{DCS}$ Peak	Peak Index	Direction of Motion
+	+	$(\lambda_u, \lambda_v)$	northeast
+	-	$(\lambda_u, -(\lambda_v + 1))$	southeast
-	+	$(-(\lambda_u + 1), \lambda_v)$	northwest
-	-	$(-(\lambda_u + 1), -(\lambda_v + 1))$	southwest

$$\text{DST}_\beta\{x\} \triangleq \frac{2C(k)}{N} \sum_{n=0}^{N-1} x(n) \sin \frac{k\pi}{N} \left( n + \beta + \frac{1}{2} \right) \quad (24)$$

are the DCT and DST with  $\alpha$  and  $\beta$  shifts in their kernels, respectively. Here,  $d$  is the shift amount and  $T$  is the sampling interval, but  $d/T$  is not necessarily an integer.

### B. Two-Dimensional Image Model

Consider a moving object casting a continuous intensity profile  $I_t(u, v)$  on a camera plane of the continuous coordinate  $(u, v)$  where the subscript  $t$  denotes the frame number. This intensity profile is then digitized on the fixed sampling grid of the camera with a sampling distance  $d$  to generate the current frame of pixels  $x_t(m, n)$  shown in Fig. 3(a) where  $m$  and  $n$  are integers. Further assume that the displacement of the object between the frames  $t-1$  and  $t$  is  $(d_u, d_v)$  such that  $I_t(u, v) = I_{t-1}(u - d_u, v - d_v)$  where  $d_u = (m_u + \nu_u)d = \lambda_u d$  and  $d_v = (m_v + \nu_v)d = \lambda_v d$ . Here,  $m_u$  and  $m_v$  are the integer components of the displacement, and  $\nu_u$  and  $\nu_v \in (-1/2, 1/2]$ . Therefore

$$\begin{aligned} x_t(m, n) &= I_t(md, nd) = I_{t-1}(md - d_u, nd - d_v) \\ x_{t-1}(m, n) &= I_{t-1}(md, nd) \end{aligned}$$

as in Fig. 3(b). Unlike the case of integer-pel movement, the displacement is not necessarily multiples of the sampling distance  $d$ . In other words,  $\nu_u$  and  $\nu_v$  do not necessarily equal zero.

For integer-pel displacements, i.e.,  $\lambda_u = m_u$  and  $\lambda_v = m_v$ , the pseudophases are computed by solving the *pseudophase motion equation* at  $(k, l)$  [19], [20]

$$\mathbf{Z}_{t-1}(k, l) \cdot \vec{\theta}_{m_u, m_v}(k, l) = \vec{x}_t(k, l), \quad \text{for } k, l \in \mathcal{N} \quad (25)$$

where  $\mathcal{N} = \{1, \dots, N-1\}$ ,  $\vec{\theta}_{m_u, m_v}$  is the pseudophase vector, and the  $4 \times 4$  system matrix  $\mathbf{Z}_{t-1}$  and the vector  $\vec{x}_t$  are composed from the 2-D-DCT-II of  $x_{t-1}(m, n)$  and the 2-D-DCT-I of  $x_t(m, n)$ , respectively:

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

$$\vec{x}_t(k, l) = \begin{bmatrix} X_t^{cc}(k, l) \\ X_t^{cs}(k, l) \\ X_t^{sc}(k, l) \\ X_t^{ss}(k, l) \end{bmatrix}, \quad \vec{\theta}_{m_u, m_v}(k, l) = \begin{bmatrix} g_{m_u, m_v}^{CC}(k, l) \\ g_{m_u, m_v}^{CS}(k, l) \\ g_{m_u, m_v}^{SC}(k, l) \\ g_{m_u, m_v}^{SS}(k, l) \end{bmatrix}.$$

Here, the 2-D-DCT-I's of  $x_{t-1}(m, n)$  are defined as

$$\begin{aligned} Z_{t-1}^{cc}(k, l) &= \text{DCCTI}\{x_{t-1}\} \triangleq \frac{4}{N^2} C(k)C(l) \\ &\times \sum_{m, n=0}^{N-1} x_{t-1}(m, n) \cos \left[ \frac{k\pi}{N}(m) \right] \cos \left[ \frac{l\pi}{N}(n) \right], \\ k, l &\in \{0, \dots, N\} \end{aligned} \quad (26)$$

$$\begin{aligned} Z_{t-1}^{cs}(k, l) &= \text{DCSTI}\{x_{t-1}\} \triangleq \frac{4}{N^2} C(k)C(l) \\ &\times \sum_{m, n=0}^{N-1} x_{t-1}(m, n) \cos \left[ \frac{k\pi}{N}(m) \right] \sin \left[ \frac{l\pi}{N}(n) \right], \\ k &\in \{0, \dots, N\}, l \in \{1, \dots, N-1\} \end{aligned} \quad (27)$$

$$\begin{aligned} Z_{t-1}^{sc}(k, l) &= \text{DSCTI}\{x_{t-1}\} \triangleq \frac{4}{N^2} C(k)C(l) \\ &\times \sum_{m, n=0}^{N-1} x_{t-1}(m, n) \sin \left[ \frac{k\pi}{N}(m) \right] \cos \left[ \frac{l\pi}{N}(n) \right], \\ k &\in \{1, \dots, N-1\}, l \in \{0, \dots, N\} \end{aligned} \quad (28)$$

$$\begin{aligned} Z_{t-1}^{ss}(k, l) &= \text{DSSTI}\{x_{t-1}\} \triangleq \frac{4}{N^2} C(k)C(l) \\ &\times \sum_{m, n=0}^{N-1} x_{t-1}(m, n) \sin \left[ \frac{k\pi}{N}(m) \right] \sin \left[ \frac{l\pi}{N}(n) \right], \\ k, l &\in \{1, \dots, N-1\} \end{aligned} \quad (29)$$

and the 2-D-DCT-II's of  $x_t(m, n)$  are defined as

$$\begin{aligned} X_t^{cc}(k, l) &\triangleq \frac{4}{N^2} C(k)C(l) \sum_{m, n=0}^{N-1} x_t(m, n) \\ &\times \cos \left[ \frac{k\pi}{N}(m + 0.5) \right] \cos \left[ \frac{l\pi}{N}(n + 0.5) \right], \\ k, l &\in \{0, \dots, N-1\} \end{aligned} \quad (30)$$

$$\begin{aligned} X_t^{cs}(k, l) &\triangleq \frac{4}{N^2} C(k)C(l) \sum_{m, n=0}^{N-1} x_t(m, n) \\ &\times \cos \left[ \frac{k\pi}{N}(m + 0.5) \right] \sin \left[ \frac{l\pi}{N}(n + 0.5) \right], \\ k &\in \{0, \dots, N-1\}, l \in \{1, \dots, N\} \end{aligned} \quad (31)$$

$$\begin{aligned} X_t^{sc}(k, l) &\triangleq \frac{4}{N^2} C(k)C(l) \sum_{m, n=0}^{N-1} x_t(m, n) \\ &\times \sin \left[ \frac{k\pi}{N}(m + 0.5) \right] \cos \left[ \frac{l\pi}{N}(n + 0.5) \right], \\ k &\in \{1, \dots, N\}, l \in \{0, \dots, N-1\} \end{aligned} \quad (32)$$

$$\begin{aligned} X_t^{ss}(k, l) &\triangleq \frac{4}{N^2} C(k)C(l) \sum_{m, n=0}^{N-1} x_t(m, n) \\ &\times \sin \left[ \frac{k\pi}{N}(m + 0.5) \right] \sin \left[ \frac{l\pi}{N}(n + 0.5) \right], \\ k, l &\in \{1, \dots, N\} \end{aligned} \quad (33)$$

where  $\{Z_{t-1}^{xx}; xx = cc, cs, sc, ss\}$  can be obtained from  $\{X_{t-1}^{xx}; xx = cc, cs, sc, ss\}$  by a simple rotation, as shown in (34), at the bottom of the next page, for  $k, l \in \mathcal{N}$  and  $\{X_{t-1}^{xx}; xx = cc, cs, sc, ss\}$  are computed and stored in memory in the previous encoding cycle.

However, for noninteger pel movement, we need to use (21) and (22) in Theorem 1 to derive the system equation at the subpixel level. If the Fourier transform of the continuous intensity profile  $I_t(u, v)$  is  $(\pi/d)$ -bandlimited and  $I_t(u, v) = I_{t-1}(u - d_u, v - d_v)$ , then according to Theorem 1, we can obtain the following 2-D relations:

$$\begin{aligned} X_t^{cc}(k, l) &= \frac{4}{N^2} C(k) C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m, n) \\ &\quad \times \cos \left[ \frac{k\pi}{N} \left( m + \lambda_u + \frac{1}{2} \right) \right] \\ &\quad \times \cos \left[ \frac{l\pi}{N} \left( n + \lambda_v + \frac{1}{2} \right) \right], \\ &\quad k, l \in \{0, \dots, N-1\} \end{aligned} \quad (35)$$

$$\begin{aligned} X_t^{cs}(k, l) &= \frac{4}{N^2} C(k) C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m, n) \\ &\quad \times \cos \left[ \frac{k\pi}{N} \left( m + \lambda_u + \frac{1}{2} \right) \right] \\ &\quad \times \sin \left[ \frac{l\pi}{N} \left( n + \lambda_v + \frac{1}{2} \right) \right], \\ &\quad k \in \{0, \dots, N-1\}, l \in \{1, \dots, N\} \end{aligned} \quad (36)$$

$$\begin{aligned} X_t^{sc}(k, l) &= \frac{4}{N^2} C(k) C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m, n) \\ &\quad \times \sin \left[ \frac{k\pi}{N} \left( m + \lambda_u + \frac{1}{2} \right) \right] \\ &\quad \times \cos \left[ \frac{l\pi}{N} \left( n + \lambda_v + \frac{1}{2} \right) \right], \\ &\quad k \in \{1, \dots, N\}, l \in \{0, \dots, N-1\} \end{aligned} \quad (37)$$

$$\begin{aligned} X_t^{ss}(k, l) &= \frac{4}{N^2} C(k) C(l) \sum_{m,n=0}^{N-1} x_{t-1}(m, n) \\ &\quad \times \sin \left[ \frac{k\pi}{N} \left( m + \lambda_u + \frac{1}{2} \right) \right] \\ &\quad \times \sin \left[ \frac{l\pi}{N} \left( n + \lambda_v + \frac{1}{2} \right) \right], \\ &\quad k, l \in \{1, \dots, N\}. \end{aligned} \quad (38)$$

Thus, we can obtain the *pseudophase motion equation* at the subpixel level:

$$\mathbf{Z}_{t-1}(k, l) \cdot \vec{\theta}_{\lambda_u, \lambda_v}(k, l) = \vec{\mathbf{x}}_t(k, l), \quad \text{for } k, l \in \mathcal{N} \quad (39)$$

where  $\vec{\theta}_{\lambda_u, \lambda_v}(k, l) = [g_{\lambda_u, \lambda_v}^{CC}(k, l), g_{\lambda_u, \lambda_v}^{CS}(k, l), g_{\lambda_u, \lambda_v}^{SC}(k, l), g_{\lambda_u, \lambda_v}^{SS}(k, l)]^T$ . A similar relationship between the DCT coefficients of  $x_t(m, n)$  and  $x_{t-1}(m, n)$  at the block boundary can be obtained in the same way:

$$\begin{aligned} \begin{bmatrix} Z_{t-1}^{cc}(k, l) & -Z_{t-1}^{cs}(k, l) \\ Z_{t-1}^{cs}(k, l) & Z_{t-1}^{cc}(k, l) \end{bmatrix} \begin{bmatrix} g_{\lambda_u, \lambda_v}^{CC}(k, l) \\ g_{\lambda_u, \lambda_v}^{CS}(k, l) \end{bmatrix} \\ = \begin{bmatrix} X_t^{cc}(k, l) \\ X_t^{cs}(k, l) \end{bmatrix}, \quad k = 0, l \in \mathcal{N}, \end{aligned} \quad (40)$$

$$\begin{aligned} \begin{bmatrix} Z_{t-1}^{cc}(k, l) & -Z_{t-1}^{sc}(k, l) \\ Z_{t-1}^{sc}(k, l) & Z_{t-1}^{cc}(k, l) \end{bmatrix} \begin{bmatrix} g_{\lambda_u, \lambda_v}^{CC}(k, l) \\ g_{\lambda_u, \lambda_v}^{SC}(k, l) \end{bmatrix} \\ = \begin{bmatrix} X_t^{cc}(k, l) \\ X_t^{sc}(k, l) \end{bmatrix}, \quad l = 0, k \in \mathcal{N}, \end{aligned} \quad (41)$$

$$\begin{aligned} \begin{bmatrix} Z_{t-1}^{cc}(k, l) & -Z_{t-1}^{cs}(k, l) \\ Z_{t-1}^{cs}(k, l) & Z_{t-1}^{cc}(k, l) \end{bmatrix} \begin{bmatrix} g_{\lambda_u, \lambda_v}^{SC}(k, l) \\ g_{\lambda_u, \lambda_v}^{SS}(k, l) \end{bmatrix} \\ = \begin{bmatrix} X_t^{sc}(k, l) \\ X_t^{ss}(k, l) \end{bmatrix}, \quad k = N, l \in \mathcal{N}, \end{aligned} \quad (42)$$

$$\begin{aligned} \begin{bmatrix} Z_{t-1}^{cc}(k, l) & -Z_{t-1}^{sc}(k, l) \\ Z_{t-1}^{sc}(k, l) & Z_{t-1}^{cc}(k, l) \end{bmatrix} \begin{bmatrix} g_{\lambda_u, \lambda_v}^{CS}(k, l) \\ g_{\lambda_u, \lambda_v}^{SS}(k, l) \end{bmatrix} \\ = \begin{bmatrix} X_t^{cs}(k, l) \\ X_t^{ss}(k, l) \end{bmatrix}, \quad l = N, k \in \mathcal{N}, \end{aligned} \quad (43)$$

$$Z_{t-1}^{cc}(k, l) g_{\lambda_u, \lambda_v}^{CS}(k, l) = X_t^{cs}(k, l), \quad k = 0, l = N, \quad (44)$$

$$Z_{t-1}^{cc}(k, l) g_{\lambda_u, \lambda_v}^{SC}(k, l) = X_t^{sc}(k, l), \quad k = N, l = 0. \quad (45)$$

In (39), the pseudophase vector  $\vec{\theta}_{\lambda_u, \lambda_v}(k, l)$  contains the information of the subpixel movement  $(\lambda_u, \lambda_v)$ . In an ideal situation where one rigid object is moving translationally within the block boundary without observable background and noise, we can find  $\vec{\theta}_{\lambda_u, \lambda_v}(k, l)$  explicitly in terms of  $\lambda_u$  and  $\lambda_v$  as such

$$\begin{aligned} \vec{\theta}_{\lambda_u, \lambda_v}(k, l) &= \begin{bmatrix} g_{\lambda_u, \lambda_v}^{CC}(k, l) \\ g_{\lambda_u, \lambda_v}^{CS}(k, l) \\ g_{\lambda_u, \lambda_v}^{SC}(k, l) \\ g_{\lambda_u, \lambda_v}^{SS}(k, l) \end{bmatrix} \\ &= \begin{bmatrix} \cos \frac{k\pi}{N} \left( \lambda_u + \frac{1}{2} \right) \cos \frac{l\pi}{N} \left( \lambda_v + \frac{1}{2} \right) \\ \cos \frac{k\pi}{N} \left( \lambda_u + \frac{1}{2} \right) \sin \frac{l\pi}{N} \left( \lambda_v + \frac{1}{2} \right) \\ \sin \frac{k\pi}{N} \left( \lambda_u + \frac{1}{2} \right) \cos \frac{l\pi}{N} \left( \lambda_v + \frac{1}{2} \right) \\ \sin \frac{k\pi}{N} \left( \lambda_u + \frac{1}{2} \right) \sin \frac{l\pi}{N} \left( \lambda_v + \frac{1}{2} \right) \end{bmatrix}. \end{aligned} \quad (46)$$

$$\begin{bmatrix} Z_{t-1}^{cc}(k, l) \\ Z_{t-1}^{cs}(k, l) \\ Z_{t-1}^{sc}(k, l) \\ Z_{t-1}^{ss}(k, l) \end{bmatrix} = \begin{bmatrix} +\cos \frac{k\pi}{2N} \cos \frac{l\pi}{2N} & +\cos \frac{k\pi}{2N} \sin \frac{l\pi}{2N} & +\sin \frac{k\pi}{2N} \cos \frac{l\pi}{2N} & \sin \frac{k\pi}{2N} \sin \frac{l\pi}{2N} \\ -\cos \frac{k\pi}{2N} \sin \frac{l\pi}{2N} & +\cos \frac{k\pi}{2N} \cos \frac{l\pi}{2N} & -\sin \frac{k\pi}{2N} \sin \frac{l\pi}{2N} & \sin \frac{k\pi}{2N} \cos \frac{l\pi}{2N} \\ -\sin \frac{k\pi}{2N} \cos \frac{l\pi}{2N} & -\sin \frac{k\pi}{2N} \sin \frac{l\pi}{2N} & +\cos \frac{k\pi}{2N} \cos \frac{l\pi}{2N} & \cos \frac{k\pi}{2N} \sin \frac{l\pi}{2N} \\ +\sin \frac{k\pi}{2N} \sin \frac{l\pi}{2N} & -\sin \frac{k\pi}{2N} \cos \frac{l\pi}{2N} & -\cos \frac{k\pi}{2N} \sin \frac{l\pi}{2N} & \cos \frac{k\pi}{2N} \cos \frac{l\pi}{2N} \end{bmatrix} \begin{bmatrix} X_{t-1}^{cc}(k, l) \\ X_{t-1}^{cs}(k, l) \\ X_{t-1}^{sc}(k, l) \\ X_{t-1}^{ss}(k, l) \end{bmatrix}. \quad (34)$$

### III. SUBPEL SINUSOIDAL ORTHOGONALITY PRINCIPLES

In [19], [20], estimation of integer-pel displacements in DCT domain utilizes the sinusoidal orthogonal principles:

$$\begin{aligned} \text{IDCT} \left\{ C(k) \cos \left[ \frac{k\pi}{N} \left( n + \frac{1}{2} \right) \right] \right\} \\ \triangleq \frac{2}{N} \sum_{k=0}^{N-1} C^2(k) \cos \left[ \frac{k\pi}{N} \left( m + \frac{1}{2} \right) \right] \cos \left[ \frac{k\pi}{N} \left( n + \frac{1}{2} \right) \right] \\ = \delta(m-n) + \delta(m+n+1) \end{aligned} \quad (47)$$

$$\begin{aligned} \text{IDST} \left\{ C(k) \sin \left[ \frac{k\pi}{N} \left( n + \frac{1}{2} \right) \right] \right\} \\ \triangleq \frac{2}{N} \sum_{k=1}^N C^2(k) \sin \left[ \frac{k\pi}{N} \left( m + \frac{1}{2} \right) \right] \sin \left[ \frac{k\pi}{N} \left( n + \frac{1}{2} \right) \right] \\ = \delta(m-n) - \delta(m+n+1) \end{aligned} \quad (48)$$

where  $\delta(n)$  is the discrete impulse function and  $m, n$  are integers. This is no longer valid at the subpixel level.

In (47) and (48), we replace the integer variables  $m$  and  $n$  by the real variables  $u$  and  $v$  and define

$$\bar{L}_c(u, v) \triangleq \sum_{k=0}^{N-1} C^2(k) \cos \frac{k\pi}{N} \left( u + \frac{1}{2} \right) \cos \frac{k\pi}{N} \left( v + \frac{1}{2} \right) \quad (49)$$

$$\bar{L}_s(u, v) \triangleq \sum_{k=0}^{N-1} C^2(k) \sin \frac{k\pi}{N} \left( u + \frac{1}{2} \right) \sin \frac{k\pi}{N} \left( v + \frac{1}{2} \right). \quad (50)$$

We show in the Appendix that

$$\bar{L}_c(u, v) = -\frac{1}{2} + \frac{1}{2} [\xi(u-v) + \xi(u+v+1)] \quad (51)$$

$$\begin{aligned} \bar{L}_s(u, v) = \frac{1}{2} \sin \left[ \pi \left( u + \frac{1}{2} \right) \right] \sin \left[ \pi \left( v + \frac{1}{2} \right) \right] \\ + \frac{1}{2} [\xi(u-v) - \xi(u+v+1)] \end{aligned} \quad (52)$$

where

$$\begin{aligned} \xi(x) &\triangleq \sum_{k=0}^{N-1} \cos \left( \frac{k\pi}{N} x \right) \\ &= \frac{1}{2} \left[ 1 - \cos \pi x + \sin \pi x \cdot \frac{\cos \frac{\pi x}{2N}}{\sin \frac{\pi x}{2N}} \right]. \end{aligned} \quad (53)$$

If  $(\pi x)/(2N)$  is so small that the second and higher order terms of  $(\pi x)/(2N)$  can be ignored, then  $\cos(\pi x/2N) \approx 1$ ,  $\sin(\pi x/2N) \approx (\pi x/2N)$ . Thus

$$\xi(x) \approx \frac{1}{2} [1 - \cos \pi x] + N \text{sinc}(x) \quad (54)$$

where  $\text{sinc}(x) \triangleq \sin(\pi x)/(\pi x)$ . For large  $N$ ,  $\xi(x)$  is approximately a sinc function whose largest peak can be identified easily at  $x = 0$  as depicted in Fig. 4(a), where  $\xi(x)$  closely resembles  $N \cdot \text{sinc}(x)$ , especially when  $x$  is small. The slope of  $\xi(x)$  is also plotted in Fig. 4(b), which shows the sharpness of  $\xi(x)$ .

A closer look at (51) and (52) reveals that either  $\bar{L}_c(u, v)$  or  $\bar{L}_s(u, v)$  consists of  $\xi$  functions and one extra term which is

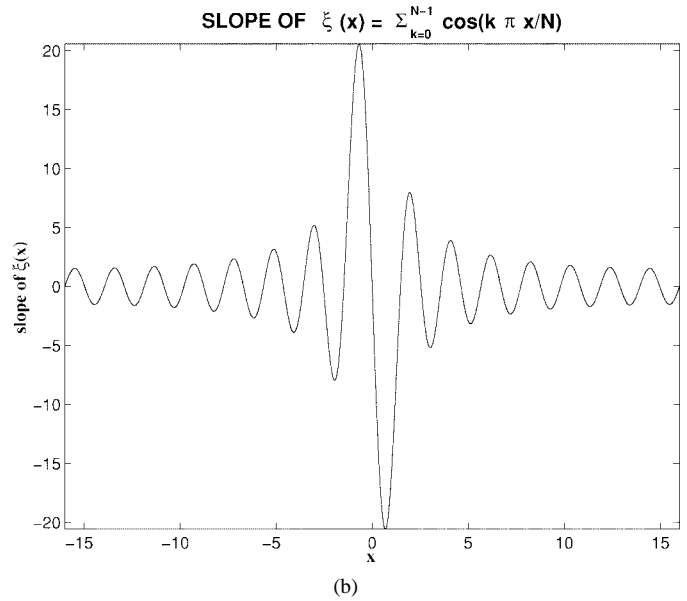
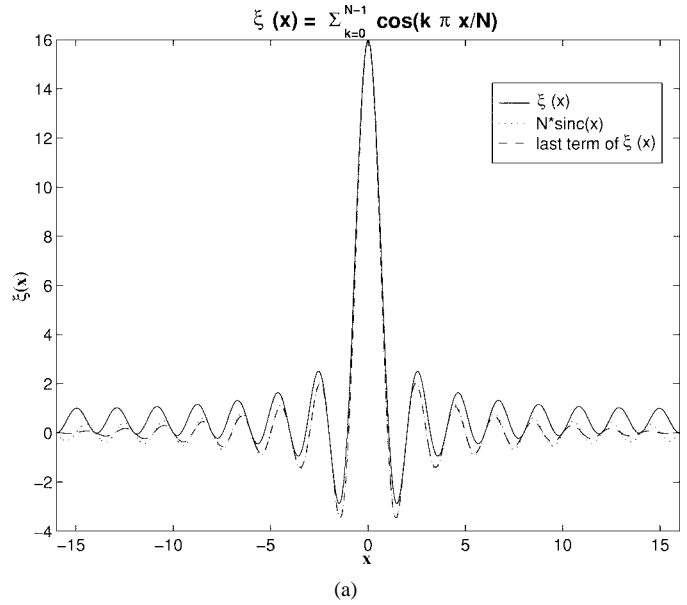


Fig. 4. Plot of  $\xi(x) = \sum_{k=0}^{N-1} \cos((k\pi/N)x)$  and its slope for  $N = 16$ . Observe the similarity between the curves of  $N \cdot \text{sinc}(x)$  and the last term of  $\xi$ .

not desirable. In order to obtain a pure form of sinc functions similar to (47) and (48), we define two modified functions  $L_c(u, v)$  and  $L_s(u, v)$  as follows:

$$L_c(u, v) \triangleq \sum_{k=0}^{N-1} \cos \frac{k\pi}{N} \left( u + \frac{1}{2} \right) \cos \frac{k\pi}{N} \left( v + \frac{1}{2} \right) \quad (55)$$

$$L_s(u, v) \triangleq \sum_{k=1}^{N-1} \sin \frac{k\pi}{N} \left( u + \frac{1}{2} \right) \sin \frac{k\pi}{N} \left( v + \frac{1}{2} \right). \quad (56)$$

Then we can show that

$$L_c(u, v) = \frac{1}{2} [\xi(u-v) + \xi(u+v+1)] \quad (57)$$

$$L_s(u, v) = \frac{1}{2} [\xi(u-v) - \xi(u+v+1)]. \quad (58)$$

Equations (55)–(58) are the equivalent form of the sinusoidal orthogonal principles (47) and (48) at the subpixel level. The

sinc functions on the right-hand side of the equations are the direct result of the rectangular window inherent in the DCT transform [34]. Fig. 5(a) and (b) illustrates  $L_s(x, -3.75)$  and  $L_c(x, -3.75)$ , respectively, where two  $\xi$  functions are interacting with each other, but their peak positions clearly indicate the displacement. However, when the displacement  $v$  is small (in the neighborhood of  $-0.5$ ),  $\xi(u-v)$  and  $\xi(u+v+1)$  move close together and addition/subtraction of  $\xi(u-v)$  and  $\xi(u+v+1)$  changes the shape of  $L_s$  and  $L_c$ . As a result, neither  $L_s$  nor  $L_c$  looks like two  $\xi$  functions, and the peak positions of  $L_s$  and  $L_c$  are different from those of  $\xi(u-v)$  and  $\xi(u+v+1)$ , as demonstrated in Fig. 5(c) and (d), respectively, where the peak positions of  $L_s(x, -0.75)$  and  $L_c(x, -0.75)$  are  $-1.25$  and  $-0.5$ , differing from the true displacement  $-0.75$ . In the extreme case,  $\xi(u-v)$  and  $\xi(u+v+1)$  cancel out each other when the displacement is  $-0.5$  such that  $L_s(x, -0.5) \equiv 0$  as shown in Fig. 5(e).

Fortunately, we can eliminate the adverse interaction of the two  $\xi$  functions by simply adding  $L_c$  and  $L_s$  together since  $L_c(x, v) + L_s(x, v) = \xi(x-v)$  as depicted in Fig. 5(f), where the sum  $L_c(x, -0.75) + L_s(x, -0.75)$  behaves like a sinc function and its peak position coincides with the displacement. Furthermore, due to the sharpness of this  $\xi$  function, we can accurately pinpoint the peak position under a noisy situation, and in turn determine the motion estimate. This property enables us to devise flexible and scalable subpixel motion estimation algorithms in the subsequent sections.

#### IV. DCT-BASED FRACTIONAL-PEL MOTION ESTIMATION

In this section, we apply the subpixel sinusoidal orthogonal principles to develop an exact subpixel motion displacement scheme without the use of interpolation to estimate half-pel and quarter-pel movements for high-quality video applications.

##### A. DCT-Based Half-Pel Motion Estimation (HDXT-ME)

From (39) in Section II, we know that the subpixel motion information is hidden, although not obvious, in the pseudophases. To obtain subpixel motion estimates, we can directly compute the pseudophases in (39), and then locate the peaks of the sinc functions after applying the subpixel sinusoidal orthogonal principles (55)–(58) to the pseudophases. Alternatively, we can have better flexibility and scalability by first using the DXT-ME algorithm to get an integer-pel motion estimate, and then utilizing the pseudophase functions  $f(k, l)$  and  $g(k, l)$  computed in the DXT-ME algorithm as in Table I to increase estimation accuracy to half pel, due to the fact that (39) has exactly the same form as (25). Specifically, based upon the subpixel sinusoidal orthogonal principles (55)–(58), the subpixel motion information can be extracted in the form of impulse functions with peak positions closely related to the displacement.

For the sake of flexibility and modularity in design and further reduction in complexity, we adopt the second approach to devise a motion estimation scheme with arbitrary fractional pel accuracy by applying the subpixel sinusoidal orthogonal principles to the pseudophase functions passed from the DXT-ME algorithm. The limitation of estimation accuracy will only

be determined by the interaction effects of the  $\xi$  functions as explained in Section III and the slope of the  $\xi$  function at and around zero, and how well the subpixel motion information is preserved in the pseudophases after sampling.

We define  $\overline{DCS}(u, v)$  and  $\overline{DSC}(u, v)$  as follows:

$$\overline{DCS}(u, v) \triangleq \sum_{k=0}^{N-1} \sum_{l=1}^{N-1} \left[ \frac{f(k, l)}{C(k)C(l)} \right] \times \cos \frac{k\pi}{N} \left( u + \frac{1}{2} \right) \sin \frac{l\pi}{N} \left( v + \frac{1}{2} \right) \quad (59)$$

$$\overline{DSC}(u, v) \triangleq \sum_{k=1}^{N-1} \sum_{l=0}^{N-1} \left[ \frac{g(k, l)}{C(k)C(l)} \right] \times \sin \frac{k\pi}{N} \left( u + \frac{1}{2} \right) \cos \frac{l\pi}{N} \left( v + \frac{1}{2} \right). \quad (60)$$

Thus, from the subpixel sinusoidal orthogonal principles (55)–(58) and the definitions of  $f(k, l)$  and  $g(k, l)$  in Table I, we can show that

$$\overline{DCS}(u, v) = \frac{1}{4} [\xi(u - \lambda_u) + \xi(u + \lambda_u + 1)] \cdot [\xi(v - \lambda_v) - \xi(v + \lambda_v + 1)] \quad (61)$$

$$\overline{DSC}(u, v) = \frac{1}{4} [\xi(u - \lambda_u) - \xi(u + \lambda_u + 1)] \cdot [\xi(v - \lambda_v) + \xi(v + \lambda_v + 1)]. \quad (62)$$

The rules to determine subpixel motion direction are summarized in Table II, and are similar to the rules in determination of integer-pel motion direction in [20].

Fig. 6 illustrates how to estimate subpixel displacements in the DCT domain. Fig. 6(c) and (d) depict the input images  $x_1(m, n)$  of size  $16 \times 16$  (i.e.,  $N = 16$ ) and  $x_2(m, n)$  displaced from  $x_1(m, n)$  by  $(2.5, -2.5)$ , respectively, at SNR = 50 dB. These two images are sampled on a rectangular grid at a sampling distance  $d = 0.625$  from the continuous intensity profile  $x_c(u, v) = \exp(-(u^2 + v^2))$  for  $u, v \in [-5, 5]$  in Fig. 6(a) whose Fourier transform is bandlimited as in Fig. 6(b) to satisfy the condition in Theorem 1. Fig. 6(e) and (f) shows the 3-D plots of the pseudophases  $f(k, l)$  and  $g(k, l)$  provided by the DXT-ME algorithm which also computes  $DCS(m, n)$  and  $DSC(m, n)$  as shown in Figs. 6(g) and (h) with peaks positioned at  $(3, 1)$  and  $(2, 2)$  corresponding to the integer-pel estimated displacement vectors  $(3, -2)$  and  $(2, -3)$ , respectively, because only the first quadrant is viewed. As a matter of fact,  $DCS(m, n)$  and  $DSC(m, n)$  have large magnitudes at  $\{(m, n); m = 2, 3, n = 1, 2\}$ .

To obtain an estimate at half-pel accuracy, we calculate  $\overline{DCS}(u, v)$  and  $\overline{DSC}(u, v)$  in (59) and (60), respectively, for  $u, v = 0 : 0.5 : N - 1$  as depicted in Fig. 6(i) and (j), where the peaks can clearly be identified at  $(2.5, 1.5)$  corresponding to the motion estimate  $(2.5, -2.5)$  exactly equal to the true displacement vector, even though the two input images do not look alike. Note that the notation  $a : r : b$  is an abbreviation of the range  $\{a + i \cdot r \text{ for } i = 0, \dots, \lfloor (b-a)/r \rfloor\} = \{a, a+r, a+2r, \dots, b-r, b\}$ . For comparison,  $\overline{DCS}(u, v)$  and  $\overline{DSC}(u, v)$  are also plotted in Fig. 6(k) and (l), respectively, for  $u, v = 0 : 0.25 : N - 1 = 0, 0.25, 0.5, \dots, N - 1.25, N - 1$  where smooth ripples are obvious due to the  $\xi$  functions

inherent in  $\overline{DCS}$  and  $\overline{DSC}$  of (61) and (62) and also have peaks at (2.5, 1.5).

Therefore, the DCT-based half-pel motion estimation algorithm (HDXT-ME) comprises three steps.

- 1) The DXT-ME algorithm estimates the integer components of the displacement as  $(\hat{m}_u, \hat{m}_v)$ .
- 2) The pseudophase functions from the DXT-ME algorithm  $f(k, l)$  and  $g(k, l)$  are used to compute  $\overline{DCS}(u, v)$  and  $\overline{DSC}(u, v)$  for  $u \in \{\hat{m}_u - 0.5, \hat{m}_u, \hat{m}_u + 0.5\}$  and  $v \in \{\hat{m}_v - 0.5, \hat{m}_v, \hat{m}_v + 0.5\}$  from (59) and (60), respectively.
- 3) Search the peak positions of  $\overline{DCS}(u, v)$  and  $\overline{DSC}(u, v)$  for the range of indexes  $\bar{\Phi} = \{(u, v) : u \in \{\hat{m}_u - 0.5, \hat{m}_u, \hat{m}_u + 0.5\}; v \in \{\hat{m}_v - 0.5, \hat{m}_v, \hat{m}_v + 0.5\}\}$  to find

$$(u_{\overline{DCS}}, v_{\overline{DCS}}) = \arg \max_{u, v \in \bar{\Phi}} |\overline{DCS}(u, v)| \quad (63)$$

$$(u_{\overline{DSC}}, v_{\overline{DSC}}) = \arg \max_{u, v \in \bar{\Phi}} |\overline{DSC}(u, v)|. \quad (64)$$

These peak positions determine the estimated displacement vector  $(\hat{\lambda}_u, \hat{\lambda}_v)$ . However, if the absolute value of  $\overline{DSC}(u, v)$  is less than a preset threshold  $\epsilon_D > 0$ , then  $\hat{\lambda}_u = -0.5$ . Likewise, if  $|\overline{DCS}(u, v)| < \epsilon_D$ ,  $\hat{\lambda}_v = -0.5$ . Therefore,

$$\hat{\lambda}_u = \begin{cases} u_{\overline{DSC}} = u_{\overline{DCS}}, & \text{if } |\overline{DSC}(u_{\overline{DSC}}, v_{\overline{DSC}})| > \epsilon_D \\ -0.5, & \text{if } |\overline{DSC}(u_{\overline{DSC}}, v_{\overline{DSC}})| < \epsilon_D \end{cases} \quad (65)$$

$$\hat{\lambda}_v = \begin{cases} v_{\overline{DCS}} = v_{\overline{DSC}}, & \text{if } |\overline{DCS}(u_{\overline{DCS}}, v_{\overline{DCS}})| > \epsilon_D \\ -0.5, & \text{if } |\overline{DCS}(u_{\overline{DCS}}, v_{\overline{DCS}})| < \epsilon_D. \end{cases} \quad (66)$$

In step 2), only those half-pel estimates around the integer-pel estimate  $(\hat{m}_u, \hat{m}_v)$  are considered due to the fact that the DXT-ME algorithm finds the nearest integer-pel motion estimate  $(\hat{m}_u, \hat{m}_v)$  from the subpixel displacement. This will significantly reduce the number of computations without evaluating all possible half-pel displacements.

In step 3), the use of  $\epsilon_D$  deals with the case of zero pseudophases when the displacement is  $-0.5$ . Specifically, if  $\lambda_u = -0.5$ , then  $g_{\lambda_u, \lambda_v}^{SC}(k, l) = 0$ ,  $\forall k, l$  which leads to  $g(k, l) = 0$  and  $\overline{DSC}(u, v) = 0$ . However, in a noisy situation, it is very likely that  $g(k, l)$  is not exactly zero, and thus neither is  $\overline{DSC}(u, v)$ . Therefore,  $\epsilon_D$  should be set very small, but large enough to accommodate the noisy case. In our experiment,  $\epsilon_D$  is empirically chosen to be 0.08. Similar consideration is made on  $\overline{DCS}(u, v)$  for  $\lambda_v = -0.5$ . It is also possible that the peak positions of  $\overline{DCS}(u, v)$  and  $\overline{DSC}(u, v)$  differ in the noisy circumstances. In this case, the arbitration rule used in the DXT-ME algorithm may be applied [19], [20].

To demonstrate the accuracy of this HDXT-ME algorithm, we use a  $16 \times 16$  dot image  $x_1$  in Fig. 7(a) as input, and displace  $x_1$  to generate the second input image  $x_2$  according to the true motion field  $\{(\lambda_u, \lambda_v) : \lambda_u, \lambda_v = -5 : 0.5 : 4\}$  shown in Fig. 7(b) through the bilinear interpolating function specified in the MPEG standard [29] which interpolates the

value  $x(m+u, n+v)$  from four neighboring pixel values for  $m, n$  being integers and  $u, v \in [0, 1)$  in the following way:

$$\begin{aligned} x(m+u, n+v) = & (1-u) \cdot (1-v) \cdot x(m, n) \\ & + (1-u) \cdot v \cdot x(m, n+1) \\ & + u \cdot (1-v) \cdot x(m+1, n) \\ & + u \cdot v \cdot x(m+1, n+1). \end{aligned} \quad (67)$$

Fig. 7(c) shows the estimated motion field by the HDXT-ME algorithm which is exactly the same as the true motion field.

Fig. 8(a)–(c) further illustrates estimation accuracy for half-pel motion estimation schemes using peak information from  $L_s(u, v)$ ,  $L_c(u, v)$ , and  $L_c(u, v) + L_s(u, v)$ , respectively. In Fig. 8(a), the “+” line indicates peak positions of  $L_s(u, v)$  found in the index range  $\{0 : 0.5 : 15\}$  for a block size  $N = 16$  with respect to different true displacement values  $\{-7 : 0.5 : 7\}$ . The “o” line specifies the final estimates after determination of motion directions from the peak signs of  $L_s(u, v)$  according to the rules in Table II. These estimates are shown to align with the reference line  $u = v$ , implying their correctness. For the true displacement  $= -0.5$ ,  $L_s(-0.5, v) \equiv 0$  for all  $v$  and  $\epsilon_D$  is used to decide whether the estimate should be set to  $-0.5$ . In Fig. 8(b),  $L_c(u, v)$  is used instead of  $L_s(u, v)$ , but  $L_c(u, v)$  is always positive, inferring that no peak sign can be exploited to determine motion direction. In Fig. 8(c),  $L_c(u, v) + L_s(u, v)$  provides accurate estimates without adjustment for all true displacement values but the index range must include negative indexes, i.e.,  $[-15 : 0.5 : 15]$ .

In the HDXT-ME algorithm, step 2) involves only nine  $\overline{DCS}(u, v)$  and  $\overline{DSC}(u, v)$  values at and around  $(\hat{m}_u, \hat{m}_v)$ . Since  $\overline{DCS}(u, v)$  and  $\overline{DSC}(u, v)$  are variants of inverse 2-D-DCT-II, the parallel and fully pipelined 2-D-DCT lattice structure proposed in [7], [26], and [27] can be used to compute  $\overline{DCS}(u, v)$  and  $\overline{DSC}(u, v)$  at a cost of  $O(N)$  operations in  $N$  steps. Furthermore, the searching in step 3) requires  $O(N^2)$  operations for one step. Thus, the computational complexity of the HDXT-ME algorithm is  $O(N^2)$  in total.

## B. DCT-Based Quarter-Pel Motion Estimation (QDXT-ME and Q4DXT-ME)

In Section III, we mention that the interaction of two  $\xi$  functions in  $L_c(u, v)$  and  $L_s(u, v)$  from (51) and (52) disassociates the peak locations with the displacement  $(\lambda_u, \lambda_v)$  for  $\lambda_u, \lambda_v \in [-1.5, 0.5]$ . In spite of this, in the HDXT-ME algorithm, we can still accurately estimate half-pel displacements by locating the peaks of  $L_s(\lambda, v)$  for true displacements  $\lambda = -N + 1 : 0.5 : N - 1$  and indexes  $v = 0 : 0.5 : N - 1$  if  $\epsilon_D$  is introduced to deal with the case for  $\lambda = -0.5$ . However, at the quarter-pel level, it does cause estimation errors around  $\lambda = -0.5$  as indicated in Fig. 8(d), where the peaks of  $L_s(\lambda, v)$  stay at  $v = 0$  for true displacements  $\lambda$  varying over  $[-1, 0]$ . As mentioned in Section III, the sum of  $L_c(\lambda, v)$  and  $L_s(\lambda, v)$  is a pure  $\xi$  function, and thus the adverse interaction is eliminated. As a result, the peak position of this sum can be used to predict precisely the displacement at either half-pel level or quarter-pel level as demonstrated in Fig. 8(c) and (f), respectively. However, for two-dimensional images,

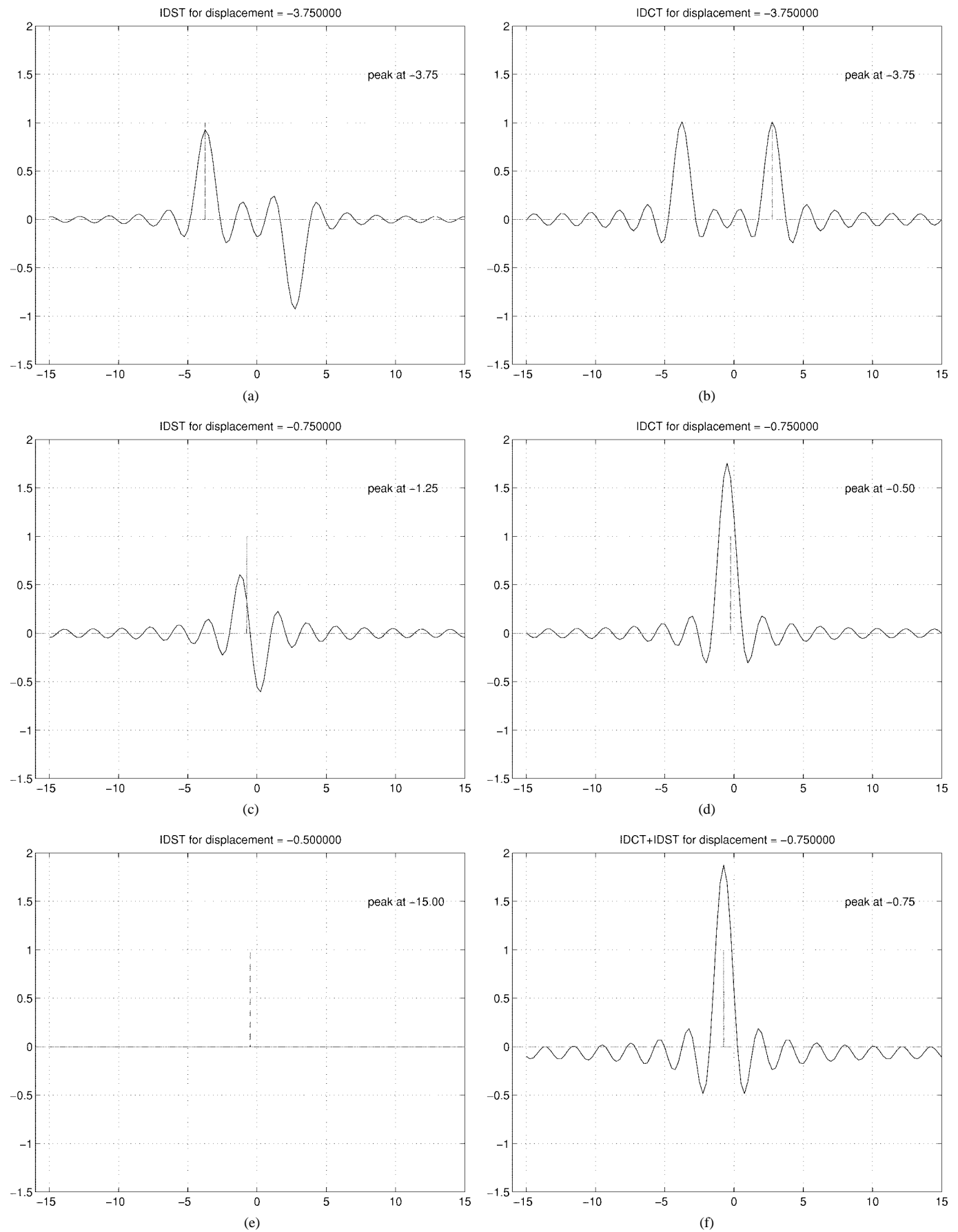


Fig. 5. Illustration of sinusoidal orthogonal principles at the subpixel level for different displacements.

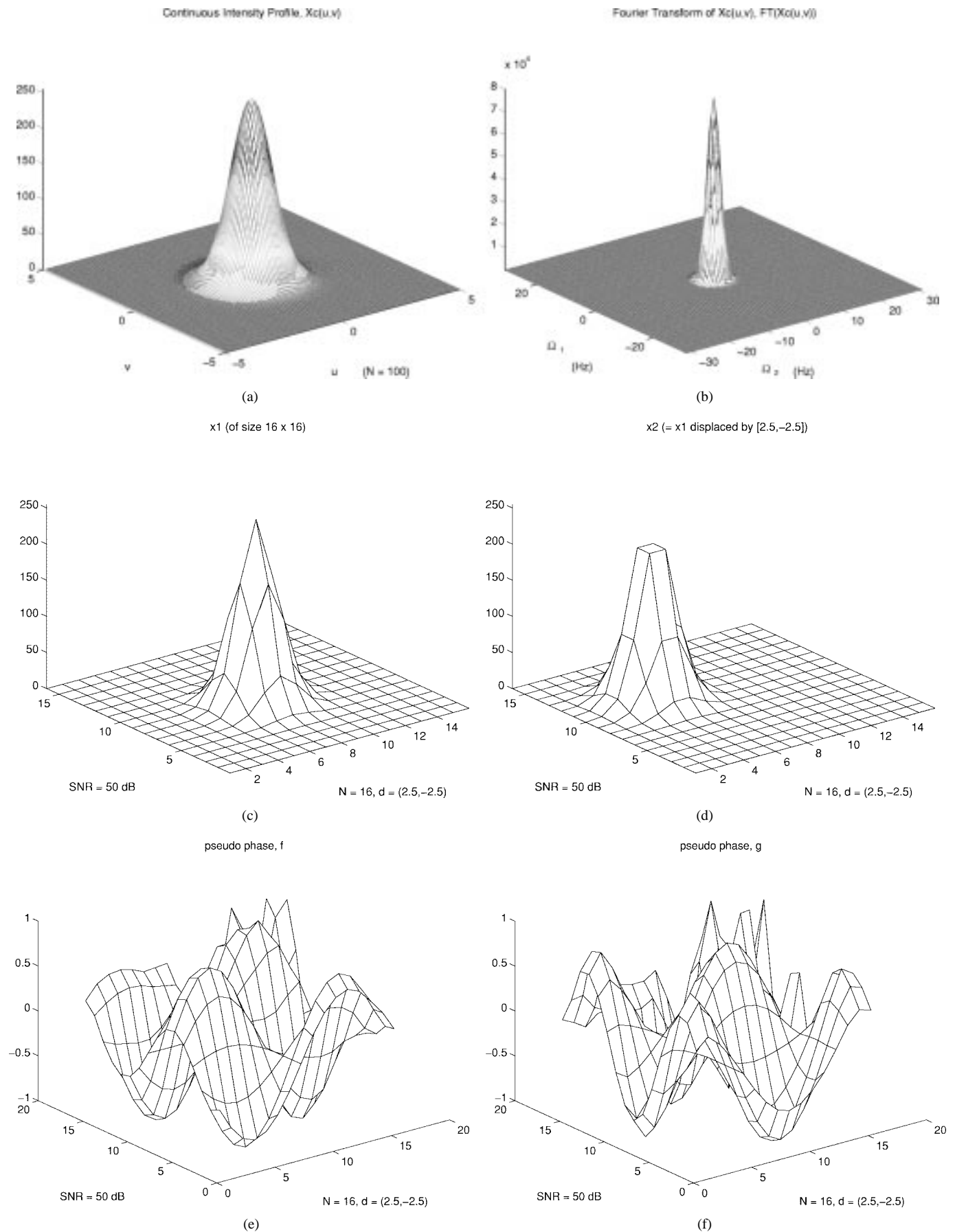


Fig. 6. Illustration of DCT-based half-pel motion estimation algorithm (HDXT-ME).

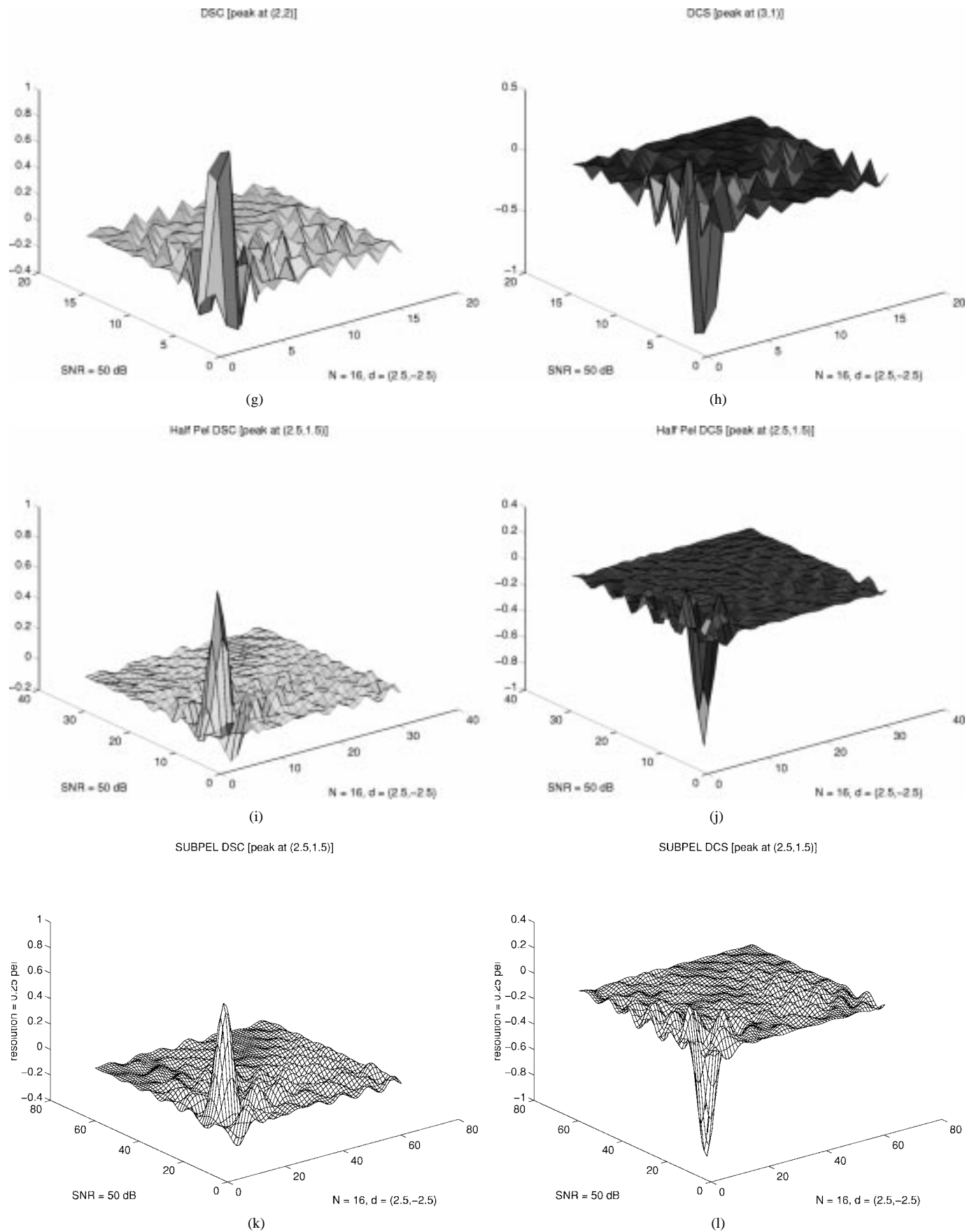


Fig. 6. (Continued.) Illustration of DCT-based half-pel motion estimation algorithm (HDXT-ME).

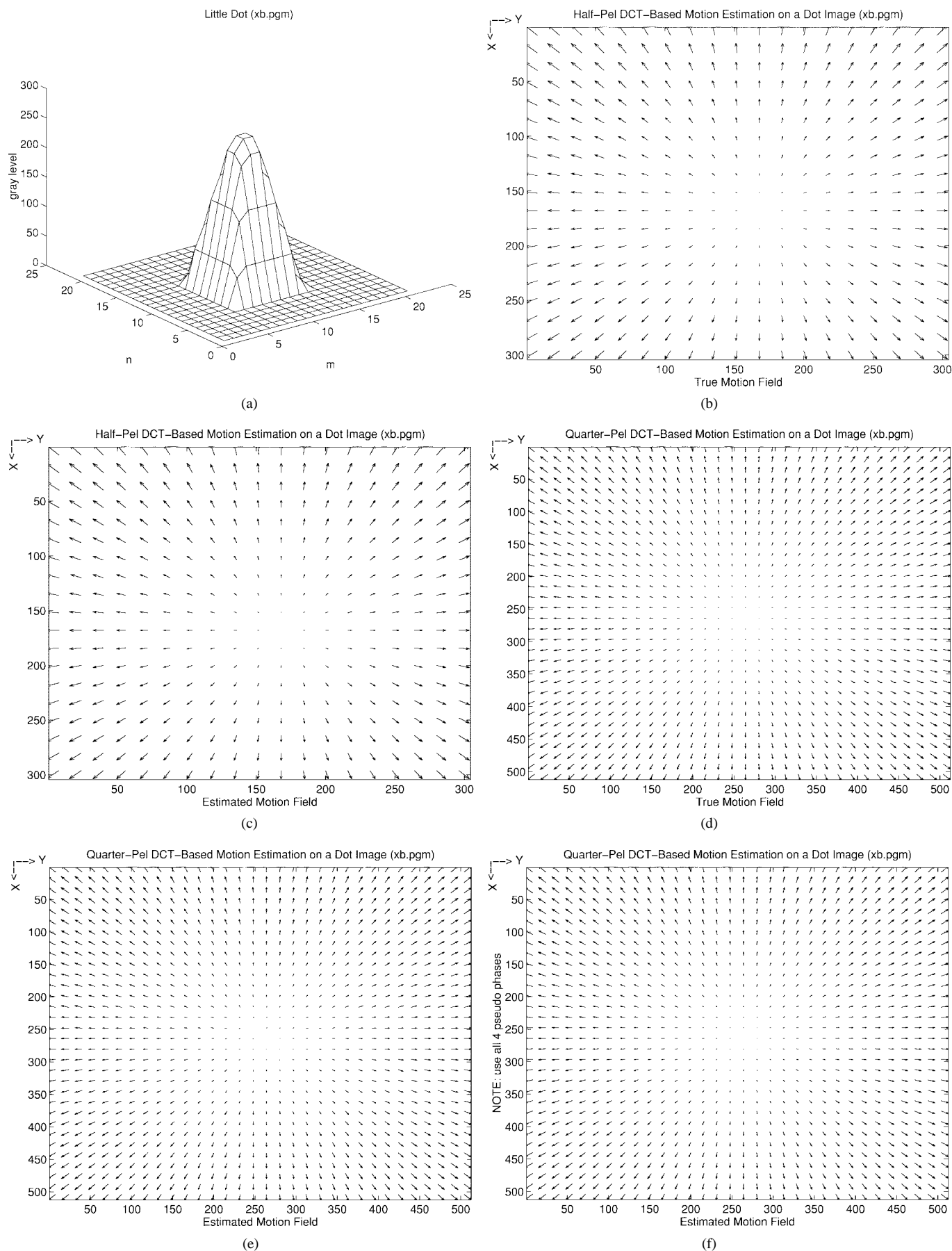


Fig. 7. Estimated motion fields (c), (e) of HDXT-ME and QDXT-ME by moving a dot image (a) according to the true motion fields (b), (d).

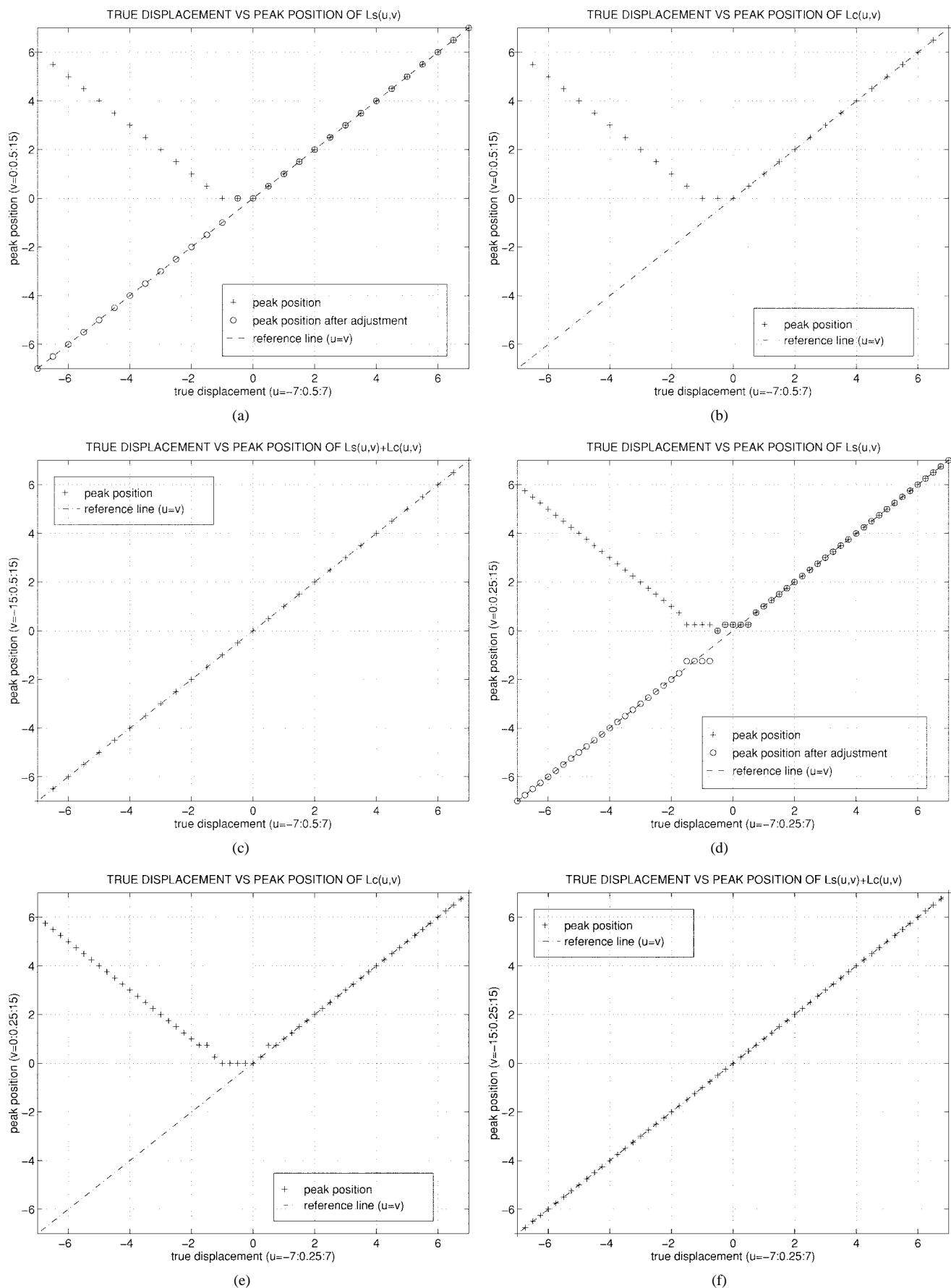


Fig. 8. Relation between true displacements and peak positions for half-pel and quarter-pel estimation. The signs of peak values in  $L_s(u, v)$  indicate the motion directions, and are used to adjust the peak positions for motion estimates.

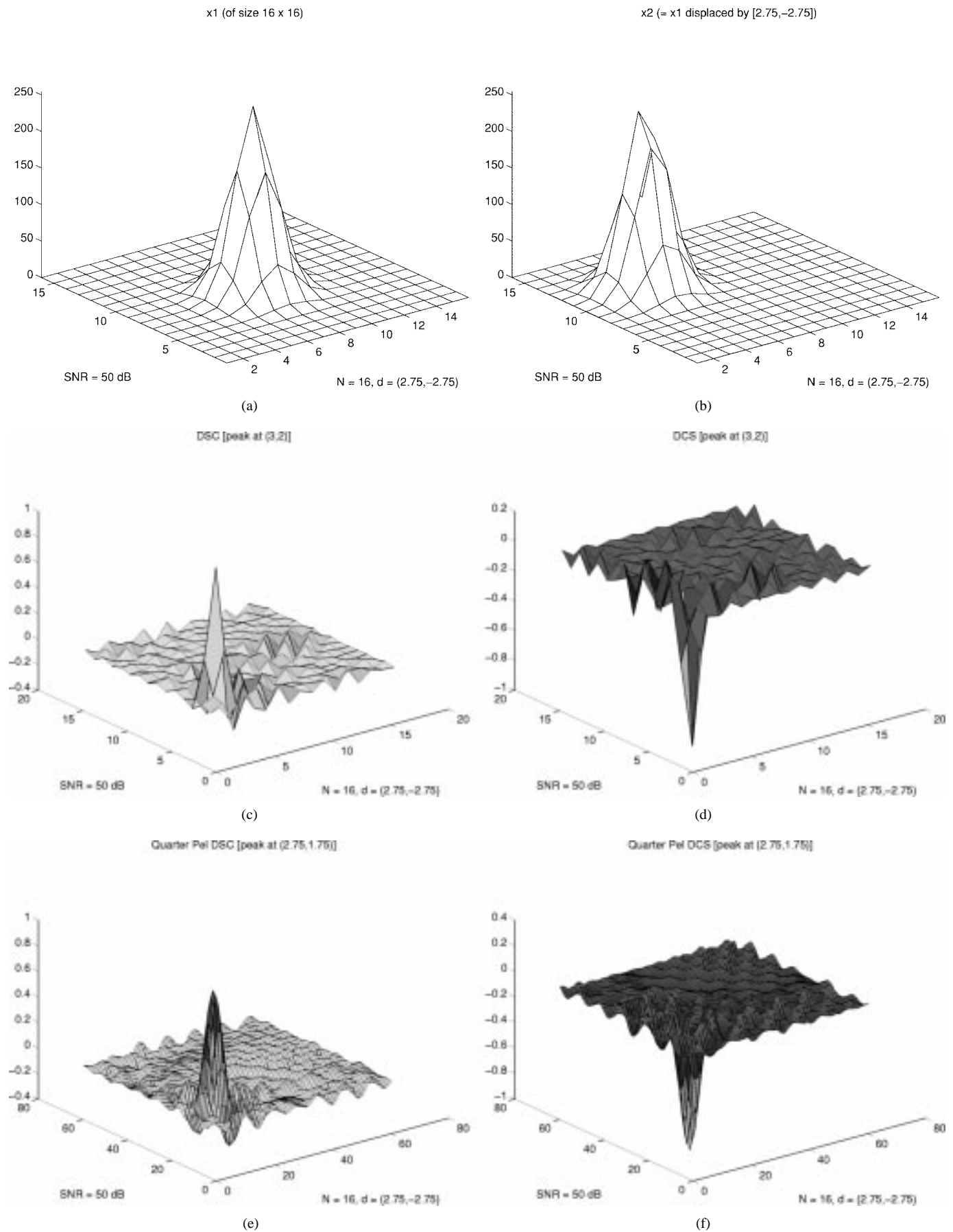


Fig. 9. Illustration of DCT-based quarter-pel motion estimation algorithm (QDXT-ME).

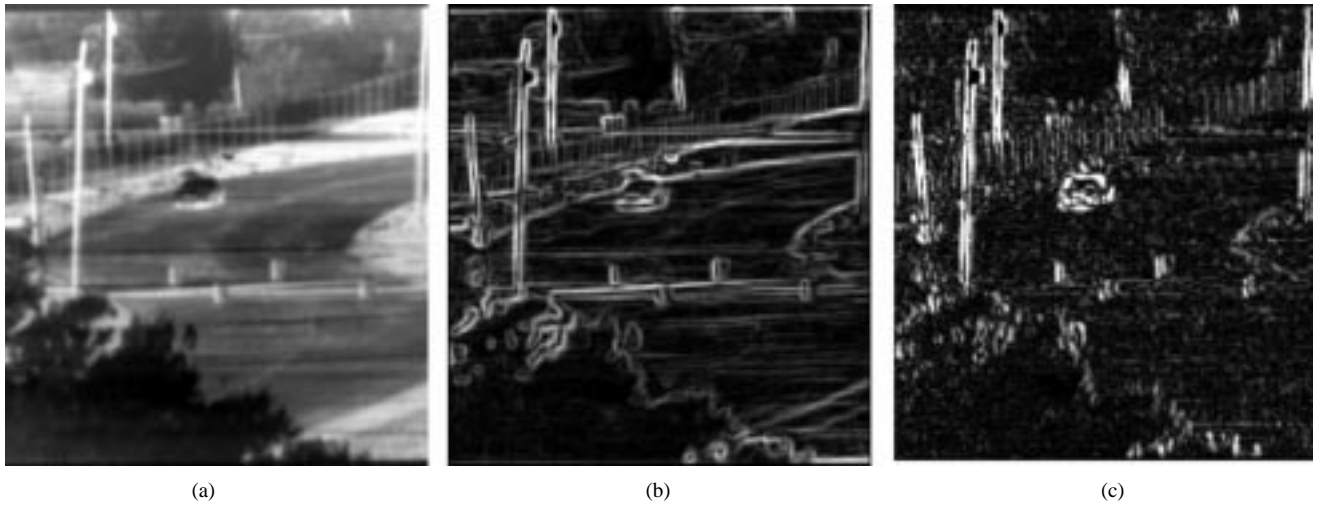


Fig. 10. Comparison of different approaches on frame 10 of “Infrared Car” sequence (HCA) for block size  $16 \times 16$  and search size  $32 \times 32$ . Visualization is applied to (c) by setting the saturation level to 25.

$\overline{DCS}$  or  $\overline{DSC}$  has four  $\xi$  functions as in (61) or (62). Since the DXT-ME algorithm provides two pseudophase functions  $f(k, l)$  and  $g(k, l)$ , only  $\overline{DCS}$  and  $\overline{DSC}$  are available for subpixel estimation. In this case, the sum of  $\overline{DCS}$  and  $\overline{DSC}$  can only annihilate two  $\xi$  functions, leaving two  $\xi$  functions as given by

$$\begin{aligned} \overline{DCS}(u, v) + \overline{DSC}(u, v) \\ = \frac{1}{2}[\xi(u - \lambda_u)\xi(v - \lambda_v) - \xi(u + \lambda_u + 1)\xi(v + \lambda_v + 1)]. \end{aligned} \quad (68)$$

Even though this sum is not a single  $\xi$  function, the estimation error of using this sum is limited to 1/4 pixel for the worst case when true displacements are either  $-0.75$  or  $-0.25$ .

The above discussion leads to the DCT-based quarter-pel motion estimation algorithm (QDXT-ME) as follows.

- 1) The DXT-ME algorithm computes the integer-pel estimate  $(\hat{m}_u, \hat{m}_v)$ .
- 2)  $\overline{DCS}(u, v)$  and  $\overline{DSC}(u, v)$  are calculated from  $f(k, l)$  and  $g(k, l)$  in (59) and (60), respectively, for the range of indexes  $\bar{\Phi} = \{(u, v) : u = \hat{m}_u - 0.75 : 0.25 : \hat{m}_u + 0.75; v = \hat{m}_v - 0.75 : 0.25 : \hat{m}_v + 0.75\}$ .
- 3) Search the peak position of  $D_2(u, v) \triangleq \overline{DCS}(u, v) + \overline{DSC}(u, v)$  over  $\bar{\Phi}$ , i.e.,

$$(u_{D2}, v_{D2}) = \arg \max_{u, v \in \bar{\Phi}} |D_2(u, v)|. \quad (69)$$

The estimated displacement vector is obtained as follows:

$$(\hat{\lambda}_u, \hat{\lambda}_v) = \begin{cases} (u_{D2}, v_{D2}), & \text{if } |D_2(u_{D2}, v_{D2})| > \epsilon_D \\ (-0.5, -0.5), & \text{if } |D_2(u_{D2}, v_{D2})| < \epsilon_D. \end{cases} \quad (70)$$

Step 3) is based on the fact that  $|D_2(\lambda_u, \lambda_v)| = 0$  if and only if  $(\lambda_u, \lambda_v) = -0.5$ . This QDXT-ME algorithm follows the same procedure as HDXT-ME, except for the search region and using the sum of  $\overline{DCS}$  and  $\overline{DSC}$ . Therefore, QDXT-ME has the same computational complexity  $O(N^2)$  as HDXT-ME.

If we modify the DXT-ME algorithm to provide the other two pseudophase functions  $g^{CC}$  and  $g^{SS}$  in addition to  $f$  and  $g$ , we can compute  $\overline{DCC}$  and  $\overline{DSS}$  in the following way:

$$\begin{aligned} \overline{DCC}(u, v) \triangleq \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} g^{CC}(k, l) \\ \times \cos \frac{k\pi}{N} \left(u + \frac{1}{2}\right) \cos \frac{l\pi}{N} \left(v + \frac{1}{2}\right) \end{aligned} \quad (71)$$

$$\begin{aligned} \overline{DSS}(u, v) \triangleq \sum_{k=1}^{N-1} \sum_{l=1}^{N-1} g^{SS}(k, l) \\ \times \sin \frac{k\pi}{N} \left(u + \frac{1}{2}\right) \sin \frac{l\pi}{N} \left(v + \frac{1}{2}\right). \end{aligned} \quad (72)$$

Then we can show that

$$\begin{aligned} D_4(u, v) \triangleq \overline{DCC}(u, v) + \overline{DCS}(u, v) + \overline{DSC}(u, v) \\ + \overline{DSS}(u, v) = \xi(u - \lambda_u)\xi(v - \lambda_v). \end{aligned} \quad (73)$$

This sum<sup>1</sup> contains only one  $\xi$  without any negative interaction effect whose peak is sharp at  $(\lambda_u, \lambda_v)$ . This leads to another quarter-pel motion estimation algorithm (Q4DXT-ME), which can accurately estimate all displacements at the quarter-pel or even finer level.

- 1) Find the integer-pel estimate  $(\hat{m}_u, \hat{m}_v)$  by the DXT-ME algorithm.
- 2) Obtain four pseudophases  $g^{CC}$ ,  $g^{CS}$ ,  $g^{SC}$ , and  $g^{SS}$  from the modified DXT-ME algorithm. Compute  $\overline{DCS}(u, v)$ ,  $\overline{DSC}(u, v)$ ,  $\overline{DCC}(u, v)$ , and  $\overline{DSS}(u, v)$  for the range of indexes  $\bar{\Phi} = \{(u, v) : u = \hat{m}_u - 0.75 : 0.25 : \hat{m}_u + 0.75; v = \hat{m}_v - 0.75 : 0.25 : \hat{m}_v + 0.75\}$ .
- 3) Search the peak position of  $D_4(u, v)$  over  $\bar{\Phi}$ :  $(u_{D4}, v_{D4}) = \arg \max_{u, v \in \bar{\Phi}} |D_4(u, v)|$ . The estimated displacement vector is then the peak position  $(\hat{\lambda}_u, \hat{\lambda}_v) = (u_{D4}, v_{D4})$ .

<sup>1</sup>These four functions can be generated naturally at the same time using the computing algorithms and architectures in [7] and [26].

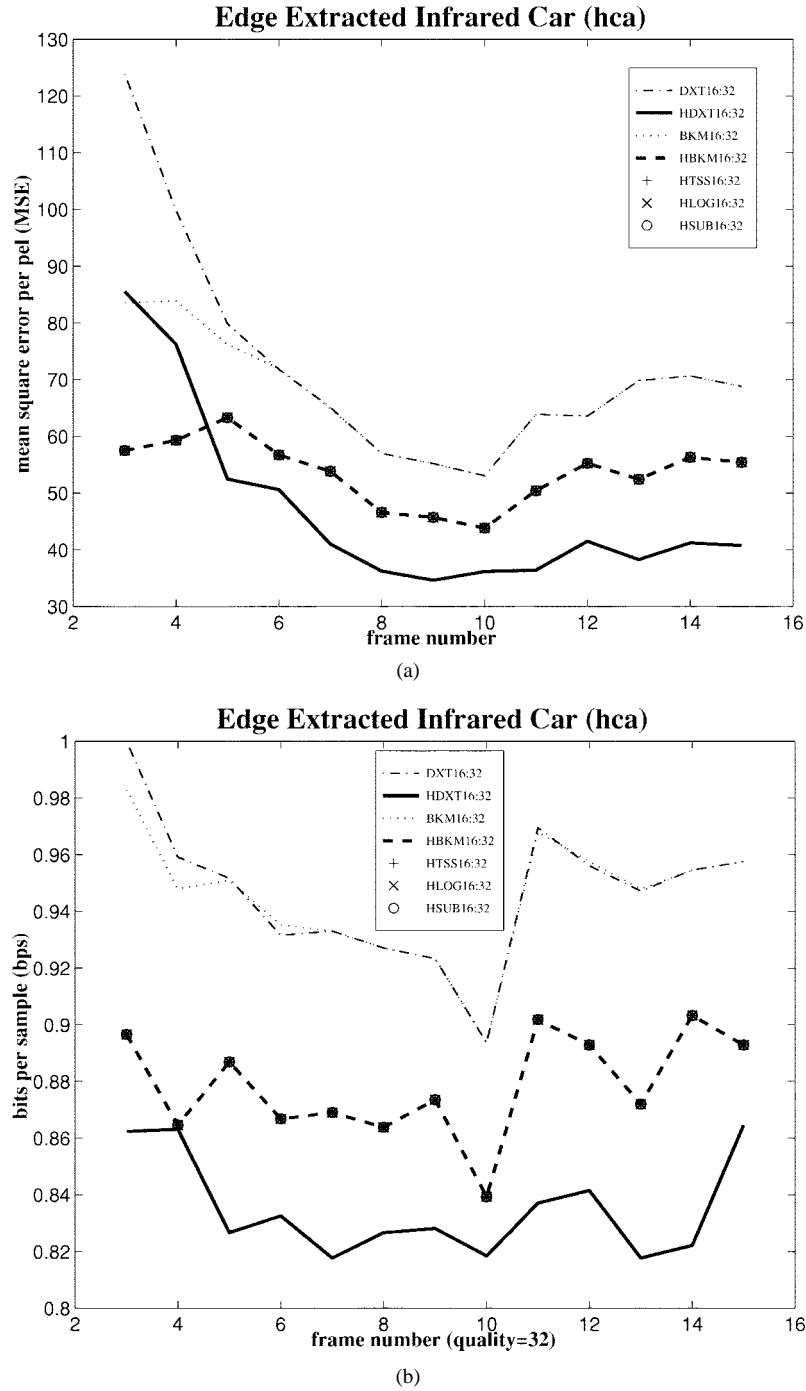


Fig. 11. Simulation results for motion estimation of half-pel accuracy on the "Infrared Car" sequence (HCA).

Fig. 9 shows the procedure to estimate a quarter-pel displacement with input images  $x_1(m, n)$  and  $x_2(m, n)$  sampled from the continuous intensity profile  $x_c(u, v)$  and its shift  $x_c(u - \lambda_u d, v - \lambda_v d)$  where  $(\lambda_u, \lambda_v) = (2.75, -2.75)$  and  $d = 0.625$  as shown in Fig. 9(a) and (b). Fig. 9(c) and (d) plots  $DSC(m, n)$  and  $DCS(m, n)$  whose peaks are both at  $(3, 2)$  corresponding to the integer-pel estimate  $(3, -3)$ . Fig. 9(e) and (f) shows the graphs of  $\overline{DSC}(u, v)$  and  $\overline{DCS}(u, v)$  at the quarter-pel level where the estimate is found to be  $(2.75, -2.75)$ .

Similar to the half-pel case, Fig. 7(e) and (f) demonstrates the accuracy of the estimated motion fields determined by

the QDXT-ME and Q4DXT-ME algorithms, respectively, as compared to the true motion field in Fig. 7(d). The first input image  $x_1(m, n)$  to both algorithms is a bandlimited dot image in Fig. 7, and the second input image  $x_2(m, n)$  is generated by shifting  $x_1(m, n)$  with respect to the true motion field in Fig. 7(d) through the bilinear interpolation. Although not obvious in the graphs, the estimates of QDXT-ME around  $-0.5$  have an estimation error up to a quarter pixel, whereas Q4DXT-ME gives us perfect estimation.

## V. SIMULATION RESULT

A set of simulations is performed on video sequences of different characteristics. Typical results can be represented by

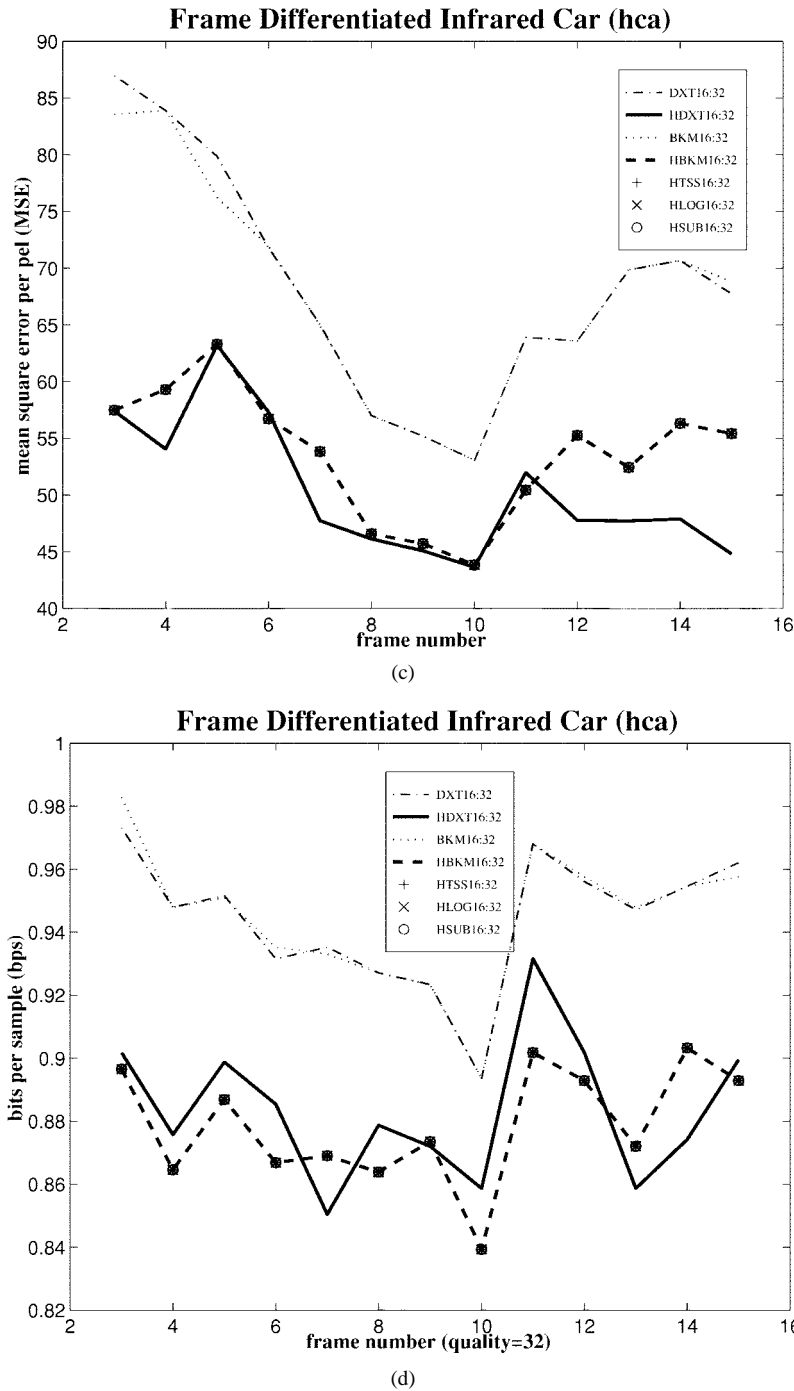


Fig. 11. (Continued.) Simulation results for motion estimation of half-pel accuracy on the “Infrared Car” sequence (HCA).

two sequences: “Miss America” (HMS) with slow head and shoulder movement accompanying with occasional eye and mouth opening, and “Infrared Car” (HCA) with a moving car viewed by a slightly shaking infrared camera. The performance of the DCT-based algorithms is compared with the full-search block-matching algorithm (BKM-ME) and its subpixel counterparts in terms of mean square error per pixel (MSE) and bits per sample (BPS). Here, MSE is defined as  $MSE = \{\sum_{m,n} [\hat{x}(m,n) - x(m,n)]^2\} / N^2$  where  $\hat{x}(m,n)$  is the reconstructed image predicted from the original image  $x(m,n)$  based upon the estimated displacement vector  $\hat{\lambda} = (\hat{\lambda}_u, \hat{\lambda}_v)$ . BPS is computed as the ratio of the total number of bits

required for a motion-compensated residual frame compressed in JPEG format to the number of pixels for each frame. For all of the MSE values computed in the experiment, the bilinear interpolation in (67) is used for comparison to reconstruct images displaced by a fractional pixel because the bilinear interpolation is used in MPEG standards for motion compensation [29], [30]. We deliberately choose to use the same spatial-domain motion compensation method for different motion estimation approaches to give us a fair comparison within the scope of this paper. Furthermore, once the motion vector is estimated, this motion estimate will be sent and used by any standard-compliant decoder which may

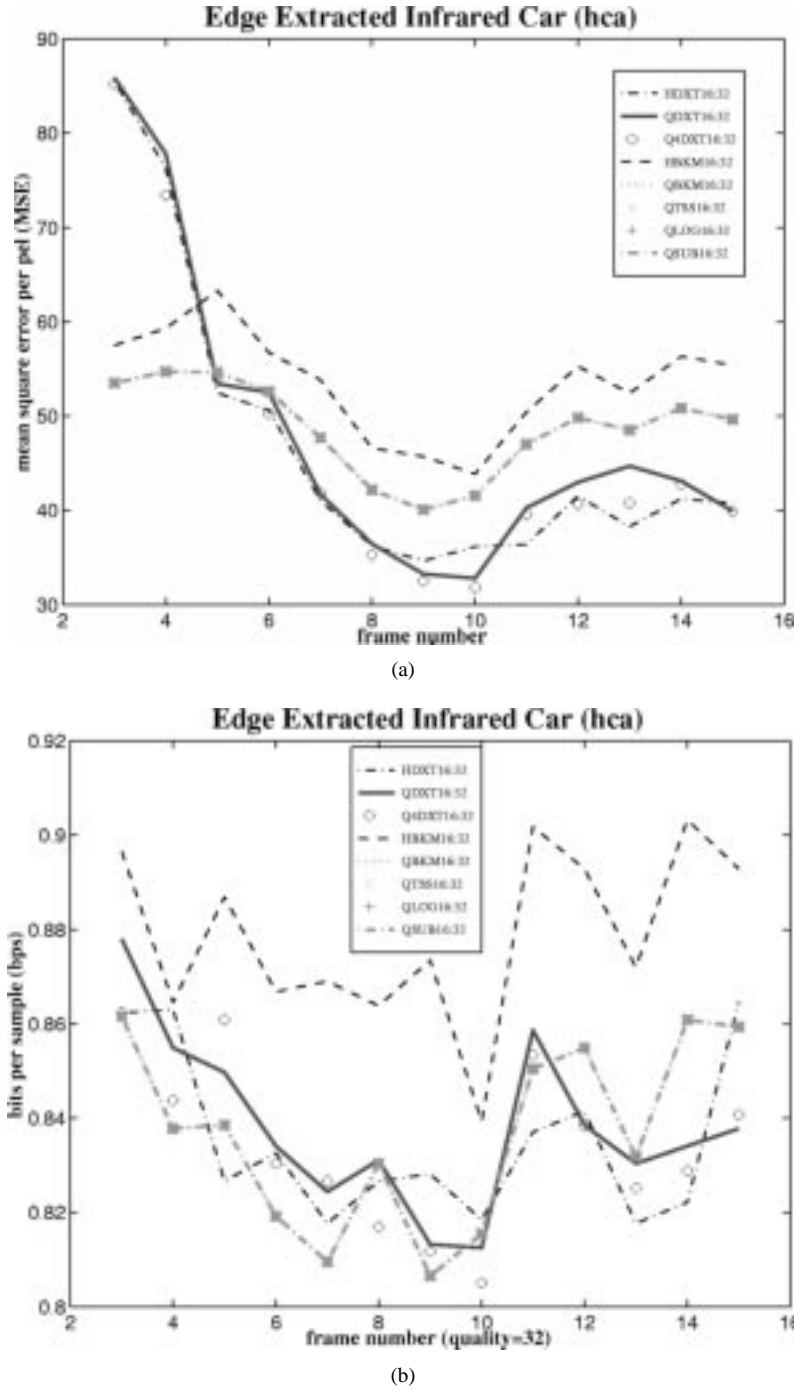


Fig. 12. Simulation results for motion estimation of quarter-pel accuracy on the “Infrared Car” sequence (HCA).

reconstruct the image by means of spatial domain methods. Therefore, it is meaningful to compare by means of the same spatial technique. For visual comparison, all residual images, generated by subtracting the original images from the reconstructed frames predicted by various motion estimation schemes, are displayed after the saturation level is reset to 25 instead of 255 to make small pixel values of the residual images be visible. In addition, the needle maps for the estimated motion fields are superimposed on the corresponding residual images.

As usual, the integer-pel BKM-ME algorithm minimizes the MAD (minimum absolute difference) function of the block

$\{x_1(m, n); m, n = 0 : 1 : N - 1\}$  over the search area  $\Phi = \{(m, n) : m, n = -(N/2) : 1 : N - 1 + (N/2)\}$  such that

$$\vec{d}_i = (\hat{n}_u, \hat{n}_v) = \arg \min_{u, v = -(N/2) : 1 : (N/2)} \sum_{m, n=0}^{N-1} |x_2(m, n) - x_1(m - u, n - v)| \times \frac{1}{N^2}. \quad (74)$$

In the simulation, two levels of subpixel block-matching motion estimation algorithms are implemented for comparison.

- 1) Half-Pel Full-Search Block-Matching Algorithm (HBKM-ME)—Similar to BKM-ME, HBKM-ME

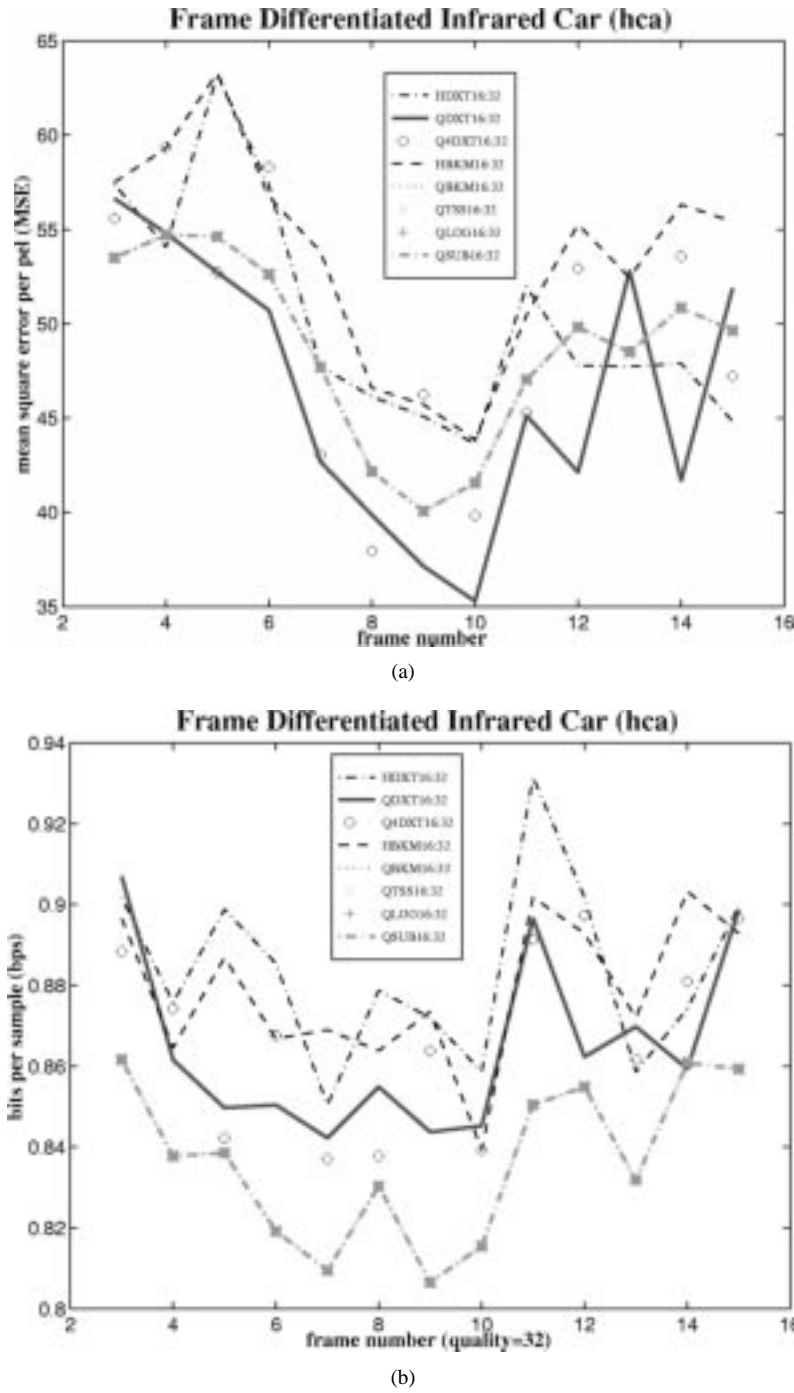


Fig. 12. (Continued.) Simulation results for motion estimation of quarter-pel accuracy on the “Infrared Car” sequence (HCA).

searches for the displacement of minimum MAD value among the integer-pel motion estimate and eight points of half-pel displacements around the integer-pel estimate as such:

$$\begin{aligned}
 \vec{d}_h &= (\hat{\lambda}_u, \hat{\lambda}_v) \\
 &= \arg \min_{u=\hat{m}_u-(1/2), \hat{m}_u, \hat{m}_u+(1/2), v=\hat{m}_v-(1/2), \hat{m}_v, \hat{m}_v+(1/2)} \\
 &\quad \sum_{m,n=0}^{N-1} |x_2(m, n) - x_1(m-u, n-v)| \\
 &\quad \times \frac{1}{N^2}. \quad (75)
 \end{aligned}$$

Searching around the integer-pel estimate instead of all possible half-pel displacements is recommended in MPEG standards to significantly reduce the overall computational complexity.

- 2) Quarter-Pel Full-Search Block-Matching Algorithm (QBKM-ME)—After the integer-pel full search block-matching (BKM-ME) motion estimation, QBKM-ME considers all half-pel and quarter-pel displacements around the integer-pel motion estimate in finding the minimum MAD value. Precisely, the estimated

TABLE III

PERFORMANCE SUMMARY OF THE 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) AND THEIR HALF-PEL (HBKM, HTSS, HLOG, HSUB) AND QUARTER-PEL (QBKM, QTSS, QLOG, QSUB) COUNTERPARTS OVER THE SEQUENCE "INFRARED CAR" (HCA); MSE DIFFERENCE IS THE DIFFERENCE FROM THE MSE VALUE OF THE 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
<b>INTEGER-PEL ACCURACY</b>						
BKM	67.902	0.000	0.0%	10156	0.945	0.0%
frame differentiated DXT	68.355	0.453	0.7%	10150	0.944	-0.1%
edge extracted DXT	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%
<b>HALF-PEL ACCURACY</b>						
HBKM	53.596	0.000	0.0%	9448	0.879	0.0%
frame differentiated HDXT	50.371	-3.224	-6.0%	9501	0.884	0.6%
edge extracted HDXT	47.013	-6.582	-12.3%	8981	0.835	-4.9%
HTSS	53.596	0.000	0.0%	9448	0.879	0.0%
HLOG	53.596	0.000	0.0%	9448	0.879	0.0%
HSUB	53.596	0.000	0.0%	9448	0.879	0.0%
<b>QUARTER-PEL ACCURACY</b>						
QBKM	48.677	0.000	0.0%	8996	0.837	0.0%
frame differentiated QDXT	46.426	-2.251	-4.6%	9298	0.865	3.4%
edge extracted QDXT	48.067	-0.611	-1.3%	9013	0.838	0.2%
frame differentiated Q4DXT	49.277	0.600	1.2%	9328	0.868	3.7%
edge extracted Q4DXT	46.769	-1.908	-3.9%	8969	0.834	-0.3%
QTSS	48.677	0.000	0.0%	8996	0.837	0.0%
QLOG	48.677	0.000	0.0%	8996	0.837	0.0%
QSUB	48.677	0.000	0.0%	8996	0.837	0.0%



Fig. 13. Comparison of different approaches on Frame 83 of "Miss America" sequence (HMS) in QCIF format for block size  $16 \times 16$  and search size  $32 \times 32$ . Visualization is applied to (c) by setting the saturation level to 25.

displacement vector is

$$\begin{aligned}
 \vec{d}_q &= (\hat{\lambda}_u, \hat{\lambda}_v) \\
 &= \arg \min_{\substack{u=\hat{m}_u-(3/4):(1/4):\hat{m}_u+(3/4), \\ v=\hat{m}_v-(3/4):(1/4):\hat{m}_v+(3/4)}} \\
 &\quad \sum_{m,n=0}^{N-1} |x_2(m,n) - x_1(m-u,n-v)| \\
 &\quad \times \frac{1}{N^2}. \quad (76)
 \end{aligned}$$

In addition to the full search block-matching approaches, we also compare with three kinds of fast-search block-matching algorithms for integer-pel, half-pel, and quarter-pel accuracy: the three-step search algorithm (TSS, HTSS, QTSS), the logarithmic search algorithm (LOG, HLOG, QLOG), and the subsampled search algorithm (SUB, HSUB, QSUB) [25]. It should be noted that all half- and quarter-pixel values for the block-matching schemes are approximated by the bilinear in-

terpolation. However, for the DCT-based subpixel algorithms, no interpolation is needed in finding the motion estimates. Therefore, the number of operations required by HBKM-ME and QBKM-ME (even for the fast search algorithms) are twice and four times as much as BKM-ME, respectively, whose computational complexity is  $O(N^4)$ , whereas the DCT-based subpixel algorithms have only a marginal increase in computations over DXT-ME of which the computational complexity is  $O(N^2)$ . In the following simulation, simple edge extraction and frame differentiation are adopted for preprocessing input images before the DCT-based algorithms to cope with the case where a block contains only part of a moving object instead of an object moving entirely inside a block, as described in detail in [19], [18]. Either preprocessing scheme adds in only  $O(N^2)$  operations as overhead, keeping the total complexity remain  $O(N^2)$ .

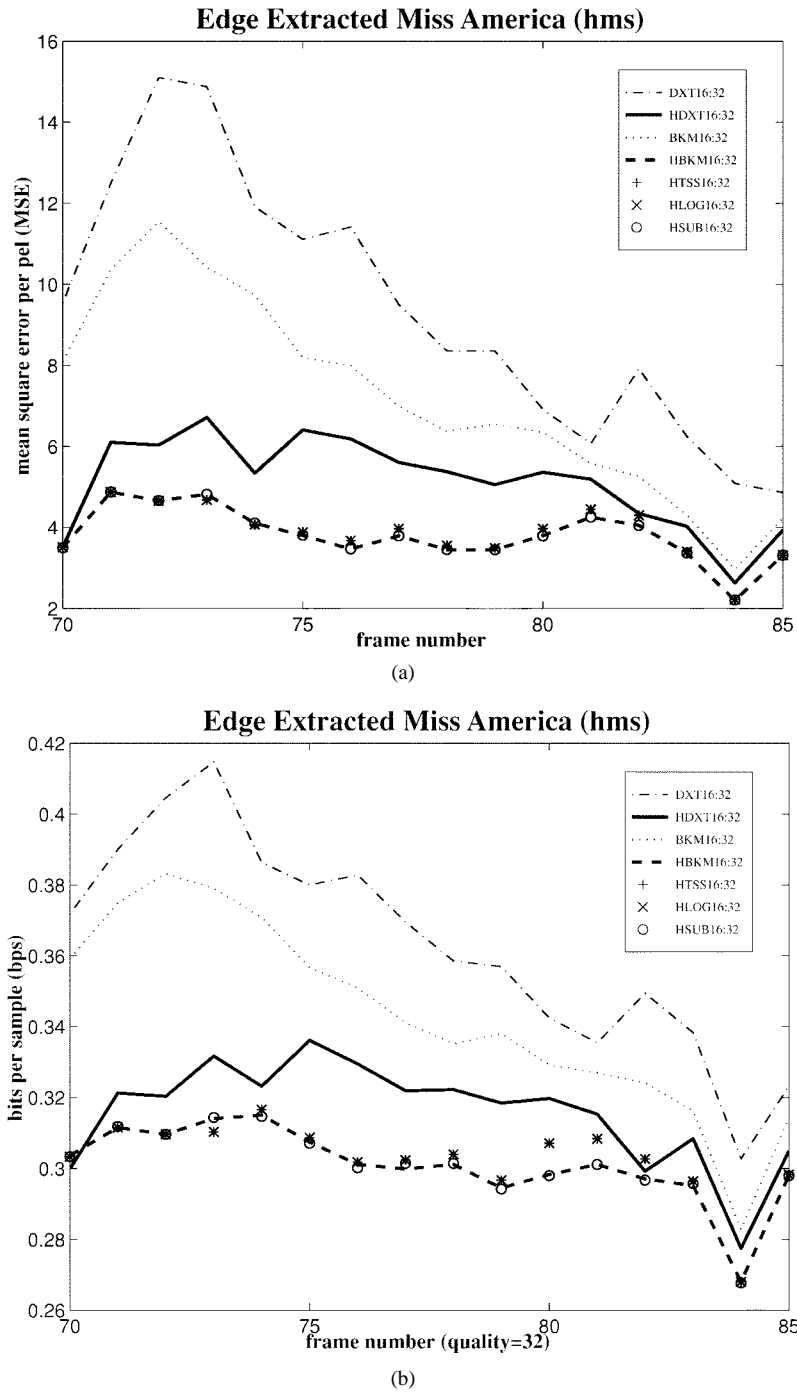


Fig. 14. Simulation results for motion estimation of half-pel accuracy on the "Miss America" sequence (HMS) in QCIF format.

Simulation is made on the "Infrared Car" sequence which has  $96 \times 112$  pixels and a moving car along the curved road viewed from a slightly shaking infrared camera. The original tenth frame is shown in Fig. 10(a), and the preprocessed tenth frames (edge extracted and frame differentiated) are shown in Fig. 10(b) and (c), respectively, where the differentiated frame contains only very small pixel values, and thus need be displayed after the visualization process; otherwise, its contents will be invisible. These small DIF values indicate only slight camera and car motion in this sequence.

The MSE and BPS values for both the block-matching approaches and the DXT-ME algorithm are plotted in Figs. 11

and 12 to compare the performances of the block-matching approaches and the DXT-ME algorithm preprocessed by frame differentiation and edge extraction with half-pel and quarter-pel accuracy, respectively. These performance curves are summarized in Table III, showing that the DXT-ME algorithm is better than the full-search block-matching algorithm for either half-pel or quarter-pel motion estimation in terms of MSE and BPS values. In Table III, the MSE difference of a motion estimation algorithm indicates how well the algorithm performs in terms of the MSE criterion in comparison to the full-search block-matching method. The MSE ratio shows this difference in percentage. Therefore, a negative MSE ratio

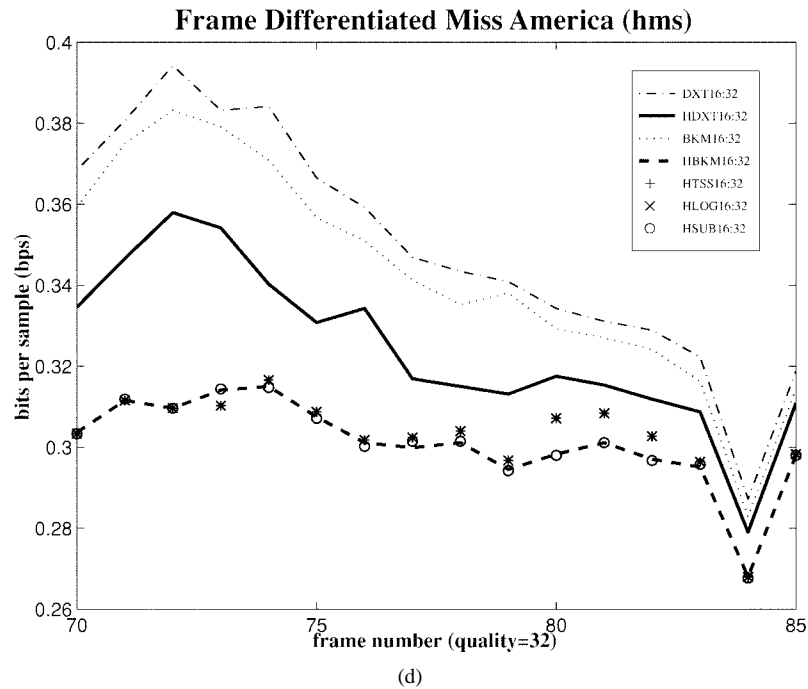
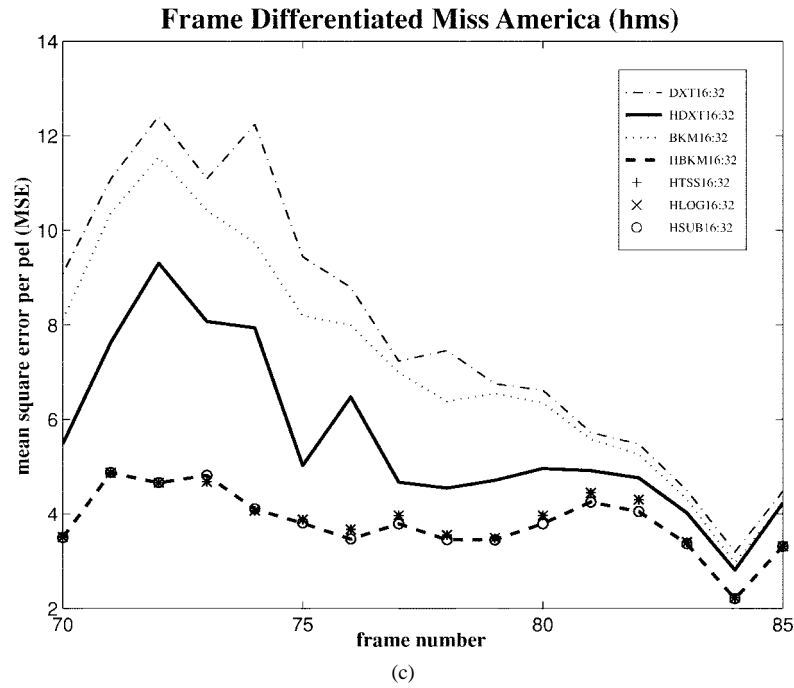


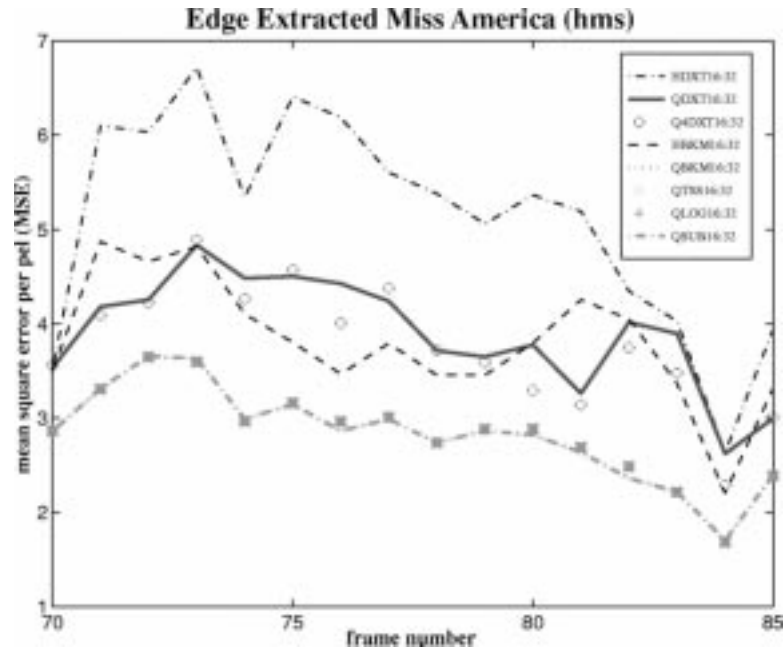
Fig. 14. (Continued.) Simulation results for motion estimation of half-pel accuracy on the "Miss America" sequence (HMS) in QCIF format.

means that the algorithm outperforms the BKM algorithm in terms of MSE. Similarly, the BPS ratio is the indicator of the performance of a motion estimation algorithm in terms of BPS. In Table III, the following observations should be noted.

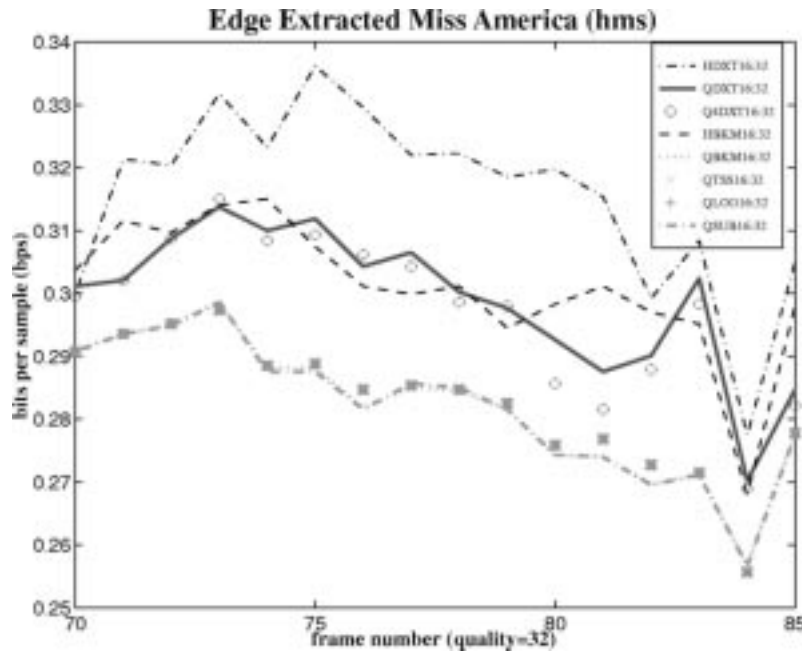
- For the integer pel accuracy, the MSE ratios for the DCT-based approaches are small positive numbers, but the MSE ratios for the DCT-based half-pel and quarter-pel algorithms are negative. Especially for the edge-extracted HDXT-ME, the MSE ratio indicates that HDXT-ME is 12.3% better than HBKM-ME in terms of MSE values.

For quarter-pel accuracy, the edge extracted Q4DXT-ME is 3.9% better than QBKM-ME. In contrast, other fast block-based subpixel motion estimation algorithms have the same MSE value as BKM.

- In terms of BPS, the full-pel frame differentiated DCT-based approach is slightly better than BKM, whereas the edge extracted half-pel and quarter-pel DCT-based approaches are better in general than the subpel BKM counterparts. In particular, the edge-extracted half-pel HDXT method is 4.9% better than the half-pel BKM approach in terms of BPS values.



(a)



(b)

Fig. 15. Simulation results for motion estimation of quarter-pel accuracy on the “Miss America” sequence (HMS) in QCIF format.

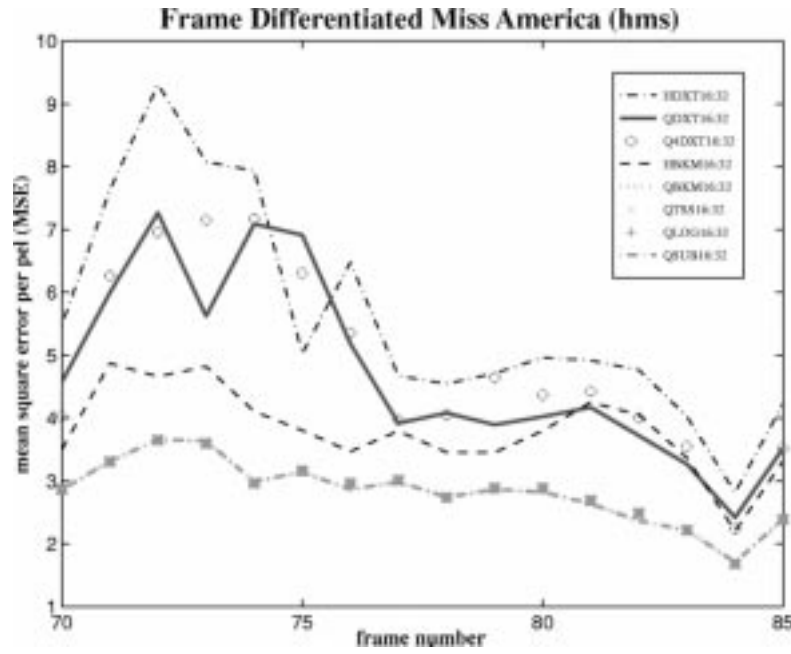
- In general, the edge-extracted subpel DCT-based approaches are better for this sequence than the frame-differentiated counterparts.

From the simulation results on the “Infrared Car” sequence, the subpel DCT-based approaches are not only better in terms of much less computational complexity, but also are better in terms of achieving smaller MSE and BPS values than the subpel block-matching approaches because of their interpolation-free characteristics.

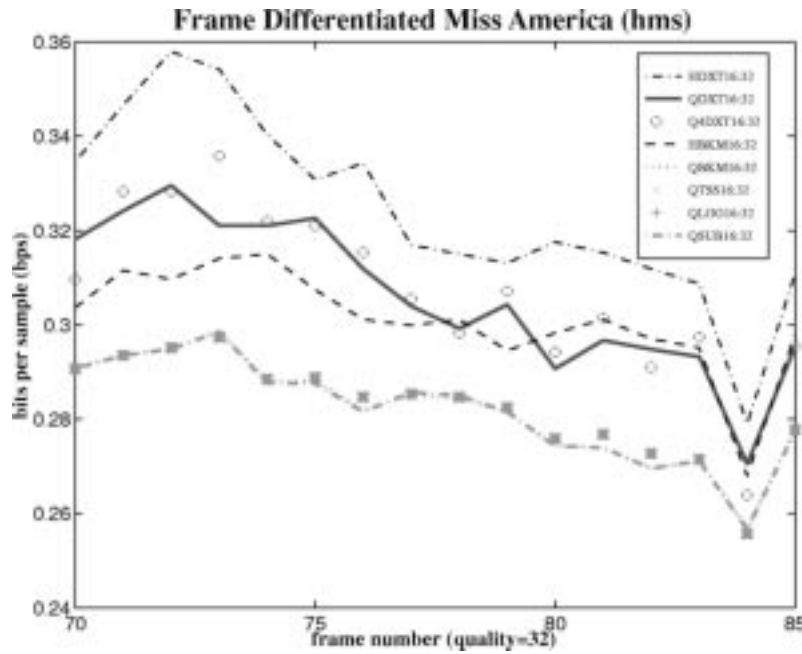
The other sequence in our simulation is “Miss America” (HMS) in QCIF format whose frame size is  $176 \times 144$ . The original frame 83 is shown in Fig. 13(a) and the preprocessed

frames in Fig. 13(b) and (c), where the differentiated frame is once again displayed after the visualization process. The white spots in Fig. 13(c) indicate only slow head and shoulder motion in this sequence.

Figs. 14 and 15 display, in terms of the MSE and BPS values, the performances of the block-matching approaches and the DXT-ME algorithm preprocessed by frame differentiation and edge extraction with half-pel and quarter-pel accuracy, respectively. These performances are summarized by averaging over the sequence in Table IV. Even though the frame-differentiated DXT-ME is better than the edge-extracted DXT-ME by achieving only 9.2% worse than BKM-ME, edge



(c)



(d)

Fig. 15. (Continued.) Simulation results for motion estimation of quarter-pel accuracy on the “Miss America” sequence (HMS) in QCIF format.

extraction seems to provide better improvement of the DXT-ME algorithm than frame differentiation for subpixel motion estimation. The coding gain from subpixel motion estimation is obvious when we compare how much improvement we can have from integer-pel accuracy to half-pel and even quarter-pel accuracy:

- HBKM-ME has 47.03% less MSE value or 12.24% less BPS value than BKM-ME, whereas QBKM-ME has 60.76% less MSE or 17.78% less BPS than BKM-ME;
- edge-extracted HDXT-ME has 45.36 % less MSE value or 12.95% less BPS value than edge-extracted DXT-ME,

whereas edge-extracted QDXT-ME has 59.79% less MSE or 18.18% less BPS.

## VI. CONCLUSION

In this paper, we develop the DCT-based subpixel motion estimation techniques based on the subpel sinusoidal orthogonal principles and preservation of subpixel motion information in DCT coefficients under the Nyquist condition. These techniques can estimate subpixel motion in the DCT domain without any interpixel interpolation at a desired level of accuracy, particularly suitable for slow object movement.

TABLE IV

PERFORMANCE SUMMARY OF THE 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) AND THEIR HALF-PEL (HBKM, HTSS, HLOG, HSUB) AND QUARTER-PEL (QBKM, QTSS, QLOG, QSUB) COUNTERPARTS OVER THE SEQUENCE "MISS AMERICA" (HMS) IN QCIF FORMAT; MSE DIFFERENCE IS THE DIFFERENCE FROM THE MSE VALUE OF THE 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
<b>INTEGER-PEL ACCURACY</b>						
BKM	7.187	0.000	0.0%	8686	0.343	0.0%
frame differentiated DXT	7.851	0.664	9.2%	8855	0.349	1.9%
edge extracted DXT	9.363	2.176	30.3%	9200	0.363	5.9%
TSS	7.862	0.675	9.4%	8910	0.352	2.6%
LOG	7.862	0.675	9.4%	8910	0.352	2.6%
SUB	7.202	0.015	0.2%	8684	0.343	0.0%
<b>HALF-PEL ACCURACY</b>						
HBKM	3.807	0.000	0.0%	7628	0.301	0.0%
frame differentiated HDXT	5.598	1.791	47.0%	8216	0.324	7.7%
edge extracted HDXT	5.116	1.308	34.4%	8000	0.316	4.9%
HTSS	3.877	0.070	1.8%	7676	0.303	0.6%
HLOG	3.877	0.070	1.8%	7676	0.303	0.6%
HSUB	3.810	0.002	0.1%	7628	0.301	0.0%
<b>QUARTER-PEL ACCURACY</b>						
QBKM	2.820	0.000	0.0%	7146	0.282	0.0%
frame differentiated QDXT	4.728	1.908	67.7%	7758	0.306	8.6%
edge extracted QDXT	3.899	1.079	38.3%	7578	0.299	6.0%
frame differentiated Q4DXT	4.874	2.054	72.8%	7785	0.307	8.9%
edge extracted Q4DXT	3.765	0.945	33.5%	7532	0.297	5.4%
QTSS	2.843	0.023	0.8%	7162	0.283	0.2%
QLOG	2.843	0.023	0.8%	7162	0.283	0.2%
QSUB	2.825	0.005	0.2%	7144	0.282	0.0%

Equally applicable to other areas as well, the proposed techniques are applied to video coding, and result in DCT-based half-pel and quarter-pel motion estimation algorithms (HDXT-ME, QDXT-ME, Q4DXT-ME) which estimate motion with half-pel or quarter-pel accuracy without interpolation of input images. This results in significant savings in computational complexity for interpolation and far less data flow compared to the conventional block-matching methods on interpolated images. Also, the resulting algorithms are more suitable for VLSI implementation [7], [27]. Furthermore, it avoids the deterioration of estimation precision caused by interpolation required in most current subpixel motion estimation schemes. In addition, the proposed DCT-based subpixel motion estimation technique and the resulting algorithms are scalable in the sense that higher estimation accuracy can be provided easily by applying the same subpel sinusoidal orthogonal principles without recomputing pseudophases. Therefore, flexible fully DCT-based codec design is possible because the same hardware can support different levels of required accuracy. Meanwhile, the computational complexity of the DCT-based algorithms is only  $O(N^2)$  compared to  $O(N^4)$  for BKM-ME or its subpixel versions. Finally, HDXT-ME, QDXT-ME, and Q4DXT-ME are DCT based, enabling us to build a low-complexity and high-throughput fully DCT-based video coder.

## APPENDIX

Equations (51)–(53) in Section III are derived as follows:

$$\begin{aligned} \bar{L}_c(u, v) &\triangleq \sum_{k=0}^{N-1} C^2(k) \cos \frac{k\pi}{N} \left(u + \frac{1}{2}\right) \cos \frac{k\pi}{N} \left(v + \frac{1}{2}\right) \\ &= -\frac{1}{2} + \sum_{k=0}^{N-1} \cos \frac{k\pi}{N} \left(u + \frac{1}{2}\right) \cos \frac{k\pi}{N} \left(v + \frac{1}{2}\right) \\ &= \frac{1}{2} \left[ -1 + \sum_{k=0}^{N-1} \cos \frac{k\pi}{N} (u - v) + \sum_{k=0}^{N-1} \cos \frac{k\pi}{N} (u + v + 1) \right] \\ &= -\frac{1}{2} + \frac{1}{2} [\xi(u - v) + \xi(u + v + 1)] \end{aligned}$$

$$\begin{aligned} \bar{L}_s(u, v) &\triangleq \sum_{k=1}^N C^2(k) \sin \frac{k\pi}{N} \left(u + \frac{1}{2}\right) \sin \frac{k\pi}{N} \left(v + \frac{1}{2}\right) \\ &= \frac{1}{2} \sin \left[ \pi \left(u + \frac{1}{2}\right) \right] \sin \left[ \pi \left(v + \frac{1}{2}\right) \right] \\ &\quad + \sum_{k=0}^{N-1} \sin \frac{k\pi}{N} \left(u + \frac{1}{2}\right) \sin \frac{k\pi}{N} \left(v + \frac{1}{2}\right) \\ &= \frac{1}{2} \sin \left[ \pi \left(u + \frac{1}{2}\right) \right] \sin \left[ \pi \left(v + \frac{1}{2}\right) \right] \\ &\quad + \frac{1}{2} \left[ \sum_{k=0}^{N-1} \cos \frac{k\pi}{N} (u - v) - \sum_{k=0}^{N-1} \cos \frac{k\pi}{N} (u + v + 1) \right] \\ &= \frac{1}{2} \sin \left[ \pi \left(u + \frac{1}{2}\right) \right] \sin \left[ \pi \left(v + \frac{1}{2}\right) \right] \\ &\quad + \frac{1}{2} [\xi(u - v) - \xi(u + v + 1)] \end{aligned}$$

$$\begin{aligned} \xi(x) &\triangleq \sum_{k=0}^{N-1} \cos \left( \frac{k\pi}{N} x \right) = \frac{1}{2} \left[ \sum_{k=0}^{N-1} e^{j(k\pi/N)x} + \sum_{k=0}^{N-1} e^{-j(k\pi/N)x} \right] \\ &= \frac{1}{2} \left[ \frac{1 - e^{j\pi x}}{1 - e^{j(\pi x/N)}} + \frac{1 - e^{-j\pi x}}{1 - e^{-j(\pi x/N)}} \right] \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2} \left[ \frac{1 - \cos \pi x - \cos \frac{\pi x}{N} + \cos \frac{\pi x}{N} (N-1)}{1 - \cos \frac{\pi x}{N}} \right] \\
&= \frac{1}{2} \left[ (1 - \cos \pi x) + \frac{\sin(\pi x) \sin\left(\frac{\pi x}{N}\right)}{1 - \cos\left(\frac{\pi x}{N}\right)} \right] \\
&= \frac{1}{2} \left[ 1 - \cos \pi x + \sin \pi x \cdot \frac{\cos \frac{\pi x}{2N}}{\sin \frac{\pi x}{2N}} \right].
\end{aligned}$$

## 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. K. Aghajan, C. D. Schaper, and T. Kailath, "Machine vision techniques for subpixel estimation of critical dimensions," *Opt. Eng.*, vol. 32, pp. 828–839, Apr. 1993.
- [3] T. Akiyama, H. Aono, K. Aoki, K. W. Ler, B. Wilson, T. Araki, T. Morishige, H. Takeno, A. Sato, S. Nakatani, and T. Senoh, "MPEG2 video codec using image compression DSP," *IEEE Trans. Consumer Electron.*, vol. 40, pp. 466–472, 1994.
- [4] B. F. Alexander and K. C. Ng, "Elimination of systematic error in subpixel accuracy centroid estimation," *Opt. Eng.*, vol. 30, pp. 1320–1331, Sept. 1991.
- [5] G. Alliance, *Grand Alliance HDTV System Specification*, Apr. 1994.
- [6] D. Brinthaup, L. Letham, V. Maheshwari, J. Othmer, R. Spiwak, B. Edwards, C. Terman, and N. Weste, "A video decoder for H.261 video teleconferencing and MPEG stored interactive video applications," in *1993 IEEE Int. Solid-State Circuits Conf.*, San Francisco, CA, 1993, pp. 34–35.
- [7] 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.
- [8] I. J. Cox, J. B. Kruskal, and D. A. Wallach, "Predicting and estimating the accuracy of a subpixel registration algorithm," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 721, Aug. 1990.
- [9] G. de Haan and W. A. C. Biezen, "Sub-pixel motion estimation with 3-D recursive search block-matching," *Signal Processing: Image Commun.*, vol. 6, pp. 229–239, June 1994.
- [10] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," *Proc. IEEE*, pp. 858–876, June 1995.
- [11] J. P. Fillard, "Subpixel accuracy location estimation from digital signals," *Opt. Eng.*, vol. 31, pp. 2465–2471, Nov. 1992.
- [12] B. Girod, "Motion-compensating prediction with fractional-pel accuracy," *IEEE Trans. Commun.*, vol. 41, p. 604, Apr. 1993.
- [13] ———, "Motion compensation: Visual aspects, accuracy, and fundamental limits," in *Motion Analysis and Image Sequence Processing*, M. I. Sezan and R. L. Lagendijk, Eds. Norwell, MA: Kluwer Academic, 1993, ch. 5.
- [14] CCITT Recommendation H.261, *Video Codec for Audiovisual Services at p × 64 kbit/s*, CCITT, Aug. 1990.
- [15] Draft CCITT Recommendation H.263, *Line Transmission of Non-Telephone Signals: Video Coding for Low Bitrate Communication*, CCITT, July 1995.
- [16] S.-L. Lu, "Comparison of motion compensation using different degrees of sub-pixel accuracy for interfield/interframe hybrid coding of HDTV image sequences," in *1992 IEEE Int. Conf. Acoust., Speech, Signal Processing*, San Francisco, CA, 1992, vol. 3, pp. 465–468.
- [17] S. P. Kim and W. Y. Su, "Subpixel accuracy image registration by spectrum cancellation," in *1993 IEEE Int. Conf. Acoust., Speech, Signal Processing*, Minneapolis, MN, 1993, vol. 5, pp. 153–156.
- [18] U.-V. Koc and K. J. R. Liu, "DCT-based motion estimation," *IEEE Trans. Image Processing*, vol. 7, pp. 948–965, July 1998.
- [19] ———, "Discrete-cosine/sine-transform based motion estimation," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Austin, TX, Nov. 1994, vol. 3, pp. 771–775.
- [20] ———, "DCT-based motion estimation," Tech. Rep. T.R.95-1, Inst. for Syst. Res., Univ. Maryland, College Park, 1995.
- [21] A. Kojima, N. Sakurai, and J. Kishigami, "Motion detection using 3D-FFT spectrum," in *IEEE ICASSP-93*, MN, Apr. 1993, vol. V, pp. V213–V216.
- [22] E. Lantz, "Subpixel signal centering and shift measurement using a recursive spectral phase algorithm," *Signal Processing*, vol. 17, p. 365, Aug. 1989.
- [23] X. Lee, "A fast feature matching algorithm of motion compensation for hierarchical video codec," in *Proc. SPIE: Visual Commun. Image Processing '92*, Boston, MA, 1992, vol. 1818, pp. 1462–1474.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] G. Madec, "Half pixel accuracy in block matching," in *Picture Coding Symp.*, Cambridge, MA, Mar. 1990.
- [29] 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.
- [30] CCITT Recommendation MPEG-2, *Generic Coding of Moving Pictures and Associated Audio*, ISO/IEC 13818, Geneva, Switzerland, 1994, H.262.
- [31] H. G. Musmann, P. Pirsch, and H.-J. Grallert, "Advances in picture coding," in *Proc. IEEE*, vol. 73, pp. 523–548, Apr. 1985.
- [32] A. N. Netravali and J. D. Robbins, "Motion compensated television coding—Part 1," *Bell Syst. Tech. J.*, vol. 58, pp. 631–670, Mar. 1979.
- [33] A. Nosratinia and M. T. Orchard, "Discrete formulation of pel-recursive motion compensation with recursive least squares updates," in *1993 IEEE Int. Conf. Acoust., Speech, Signal Processing*, Minneapolis, MN, 1993, vol. 5, pp. 229–232.
- [34] A. Papoulis, *Signal Analysis*. New York: McGraw-Hill, 1977.
- [35] P. Pirsch, N. Demassieux, and W. Gehrke, "VLSI architecture for video compression—A survey," in *Proc. IEEE*, pp. 220–246, Feb. 1995.
- [36] R. W. Schafer and L. R. Rabiner, "A digital signal processing approach to interpolation," in *Proc. IEEE*, pp. 692–702, June 1973.
- [37] M. R. Shortis, T. A. Clarke, and T. Short, "A comparison of some techniques for the subpixel location of discrete target images," in *Proc. SPIE: Videometrics III*, Boston, MA, 1994, vol. 2350, pp. 239–250.
- [38] G. A. Thomas, "Television motion measurement for DATV and other applications," Tech. Rep. 11, BBC Research Department, 1987.
- [39] Q. Tian and M. N. Huhns, "Algorithms for subpixel registration," *Comput. Vision, Graphics and Image Processing*, vol. 35, pp. 220–233, 1986.
- [40] S.-I. Uramoto, A. Takabatake, and M. Yoshimoto, "A half-pel precision motion estimation processor for NTSC-resolution video," *IEICE Trans. Electron.*, vol. 77, p. 1930, Dec. 1994.
- [41] G. A. W. West and T. A. Clarke, "A survey and examination of subpixel measurement techniques," in *Proc. SPIE: Close-Range Photogrammetry Meets Machine Vision*, Zurich, Switzerland, 1990, vol. 1395, pp. 456–463.
- [42] M. Ziegler, "Hierarchical motion estimation using the phase correlation method in 140 mbit/s HDTV-coding," in *Signal Processing of HDTV, II*, Turin, Italy, 1990, pp. 131–137.



**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 from the University of Maryland, College Park (UMCP), in 1992 and 1996, respectively.

During 1989–1990, he was a full-time Teaching Assistant at NCTU. From 1991 to 1992, he was with the Plasma Research Center at UMCP. From 1992 through 1996, he was a Research Assistant at the Institute for Systems Research and the Lab Manager of the Digital Signal Processing Lab. at UMCP. He is currently with Bell Laboratories, Lucent Technologies, Murray Hill, NJ. 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 the 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 Department of Electrical Engineering and the Institute for Systems Research, University of Maryland, College Park, where he is an Associate Professor. During his sabbatical leave in 1996–1997, he was a 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 and book chapters. He is a founder of Odyssey Technologies.

Dr. Liu 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), 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, and many others. He has been an Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING. He is a Guest Editor of Special Issues on Multimedia Signal Processing and Technology of PROCEEDINGS OF THE IEEE, a Guest Editor of the Special Issue on Signal Processing for Wireless Communications of the IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS, an Editor of *Journal of VLSI Signal Processing*, and a founding member of the Multimedia Signal Processing Technical Committee of the IEEE Signal Processing Society. He is the Series Editor of the Marcel Dekker series on signal processing. Dr. Liu is the co-editor of the books, *High Performance VLSI Signal Processing: Volume I: System Design and Methodology*; *Vol II: Algorithms, Architectures, and Applications* (IEEE Press).