DISCRETE-COSINE/SINE-TRANSFORM BASED MOTION ESTIMATION*

Ut-Va Koc and *K. J. Ray Liu*

Electrical Engineering Department and Institute for Systems Research
University of Maryland at College Park
College Park, Maryland 20742
koc@eng.umd.edu and kjrliu@eng.umd.edu

ABSTRACT

A new motion estimation scheme, Discrete-Cosine/Sine-Transform Based Motion Estimation (DXT-ME) utilizing the principle of orthogonality of cosine and sine functions to estimate, in the transform domain, displacements from the motion information contained in the *pseudo-phases* of the images of moving objects, is proposed. The computational complexity of this method is only $O(N^2) + O_{dct}$ for an image of size $N \times N$ in comparison to $O(N^4)$, the complexity of full search block matching approach, where O_{dct} is the complexity of DCT/DST coding. Furthermore, incorporation of DXT-ME with DCT-based video standards achieves further savings of computations and makes symmetric codecs feasible. Unlike pel-recursive algorithm, this scheme is not susceptible to noise. For complicated scenery or large moving objects, simple preprocessing is performed on images to extract the features of moving objects before applying DXT-ME. Simulation on some video sequences is presented to compare this scheme with the block matching method.

1. INTRODUCTION

In recent years, great interests have been found in motion estimation due to its various promising applications in multimedia, video telephony, high definition television (HDTV), automatic video tracker (AVT) and computer vision, etc. Extensive research has been done over many years in developing new algorithms and designing cost-effective and massively parallel hardware architectures suitable for current VLSI technology. Unfortunately, many currently available motion estimation schemes, such as Full Search Block Matching Algorithm (BMA-ME), and Complex Lapped Transform Approach (CLT-ME), search for candidate blocks over a larger search area and thus result in a very high computational burden on the hardware. Other estimation approaches, such as Pel-Recursive Algorithm (PRA-ME) and Optical Flow Approach (OFA-ME), are very vulnerable to noise by virtue of their involving only local operations.

In this paper, we present a novel algorithm for motion estimation, called Discrete-Cosine/Sine-Transform Based Motion Estimation (DXT-ME) to estimate motion in discrete-cosine-transform/discrete-sine-transform (DCT/DST or DXT for short) domain. DXT-ME is based on the principle of orthogonality of cosine and sine functions.

This new algorithm has certain merits over conventional methods. It has very low computational complexity (in the order of N^2 compared to N^4 for BMA-ME) and is robust even in a very noisy environment. This algorithm can also take 2D-DCT coefficients of images as input to estimate motions and therefore can be incorporated in encoders of most current video compression standards such as MPEG and CCITT H261 [1][2]. In other words, DXT-ME can combine both DCT encoders and motion estimation into a single component to achieve further saving of operations and hardware complexity. Furthermore, the transmitter and receiver structures are symmetric for this combined DCT-based motion estimation scheme.

2. PRINCIPLE OF SINUSOIDAL ORTHOGONALITY

We first assume that an object moves translationally by m_1 in X direction and n_1 in Y direction as viewed on the camera plane and within the scope of the camera in a noiseless environment. Then we can extract the motion vector out of the two consecutive frames of the images of that moving object by making use of one of these two orthogonal equations:

$$\sum_{k=0}^{N-1} D^2(k) \sin[\frac{k\pi}{N}(m+\frac{1}{2})] \sin[\frac{k\pi}{N}(n+\frac{1}{2})] \qquad (1)$$
$$= \frac{N}{2}[\delta(m-n) - \delta(m+n-1)],$$
$$\text{where } D(k) = \begin{cases} \frac{1}{2}, & \text{for } k = N, \\ 1, & \text{otherwise,} \end{cases}$$

$$\sum_{k=0}^{N-1} C^2(k) \cos[\frac{k\pi}{N}(m+\frac{1}{2})] \cos[\frac{k\pi}{N}(n+\frac{1}{2})] \qquad (2)$$
$$= \frac{N}{2}[\delta(m-n) + \delta(m+n-1)],$$
$$\text{where } C(k) = \begin{cases} \frac{1}{2}, & \text{for } k = 0, \\ 1, & \text{otherwise.} \end{cases}$$

*This work is supported in part by the ONR grant N00014-93-1-0566, NSF grant MIP9309506, and MIPS/MicroStar.

0-8186-6950-0/94 $4.00 © 1994 IEEE

As well known, Fourier transform of a shifted signal contains the information about the amount of this shift in its phase. Similarly DCT coefficients of a shifted signal do also carry this information, even though it is not clear so far how to extract the information. To facilitate explanation of the idea behind DXT-ME, let us turn to the case of one-dimensional signals. Suppose that the signal $\{x_1(n); n \in \{0, \ldots, N-1\}\}$ is right shifted by the amount m (for simplicity, $m > 0$) to generate $\{x_2(n); n \in \{0, \ldots, N-1\}\}$ and has zero values at $n \leq m$. Therefore, $x_2(n) = x_1(n-m)$ for $n \in \{m, \ldots, m+N-1\}$ and 0 elsewhere. It can be easily shown that

$$X_2^C(k) = Z_1^C(k)\cos[\frac{k\pi}{N}(m+\frac{1}{2})] - Z_1^S(k)\sin[\frac{k\pi}{N}(m+\frac{1}{2})],$$
$$X_2^S(k) = Z_1^S(k)\cos[\frac{k\pi}{N}(m+\frac{1}{2})] + Z_1^C(k)\sin[\frac{k\pi}{N}(m+\frac{1}{2})].$$

Here X_2^S and X_2^C are DST and DCT of the second kind of x_2 respectively whereas Z_1^S and Z_1^C are DST and DCT of the first kind of x_1 [3]. The shift value m is embedded solely in $\sin[\frac{k\pi}{N}(m+\frac{1}{2})]$ and $\cos[\frac{k\pi}{N}(m+\frac{1}{2})]$ named as *pseudo-phases* analogous to phases in Fourier transform of shifted signals. After solving the above equations for the pseudo-phases, we apply the orthogonal principles (Eqs. 1 and 2) for DCT/DST kernels, to the pseudo-phases and can then determine the sign and magnitude of m from signs and locations of peaks of the δ functions.

3. THE ALGORITHM OF DXT-ME

The above idea for 1-D signals can be extended to 2-dimensional images. As depicted in Fig. 1, images of previous and current frames, x_{t-1} and x_t, are fed into 2D-DCT-II and 2D-DCT-I encoders respectively. 2D-DCT-II is simply an extension of one dimensional DCT-DST transforms of the second kind (1D-DCT/DST-II) [3] and consists of four types of coefficients, DCCTII, DCSTII, DSCTII, and DSSTII, each of which is defined by a two-dimensional separable function using 1D-DCT/DST-II kernels. For example, DCCTII and DSSTII coefficients can be computed as follows:

$$X_t^{cc}(k,l) = \frac{4}{N^2}C(k)C(l)\sum_{m,n=0}^{N-1} x_t(m,n) \cdot \cos[\frac{k\pi}{N}(m+0.5)]\cos[\frac{l\pi}{N}(n+0.5)]; \quad k,l \in \{0,\ldots,N-1\},$$
$$X_t^{ss}(k,l) = \frac{4}{N^2}D(k)D(l)\sum_{m,n=0}^{N-1} x_t(m,n) \cdot \sin[\frac{k\pi}{N}(m+0.5)]\sin[\frac{l\pi}{N}(n+0.5)]; \quad k,l \in \{1,\ldots,N\}.$$

Other types of 2D-DCT-II coefficients can be obtained in a similar way. In the same fashion, the 2-D DCT-DST coefficients of the first kind (2D-DCT-I) are computed

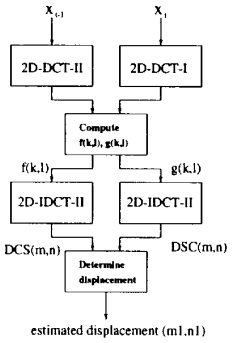

Figure 1: Block Diagram of the Algorithm, DXT-ME

by means of one dimensional DCT/DST kernels of the first kind defined as:

$$\text{1D-DCT-I kernel} = \frac{2}{N}C(k)\cos[\frac{k\pi}{N}(m)];$$
$$\text{1D-DST-I kernel} = \frac{2}{N}D(k)\sin[\frac{k\pi}{N}(m)].$$

After the calculation of the coefficients, pseudo-phase functions denoted as $f(\cdot,\cdot)$ and $g(\cdot,\cdot)$ are computed by solving the following 4×4 linear equations:

$$\begin{bmatrix} Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{cs}(k,l) & -Z_{t-1}^{sc}(k,l) & Z_{t-1}^{ss}(k,l) \\ Z_{t-1}^{cs}(k,l) & Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{ss}(k,l) & -Z_{t-1}^{sc}(k,l) \\ Z_{t-1}^{sc}(k,l) & -Z_{t-1}^{ss}(k,l) & Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{cs}(k,l) \\ Z_{t-1}^{ss}(k,l) & Z_{t-1}^{sc}(k,l) & Z_{t-1}^{cs}(k,l) & Z_{t-1}^{cc}(k,l) \end{bmatrix} \cdot \begin{bmatrix} CC(k,l) \\ CS(k,l) \\ SC(k,l) \\ SS(k,l) \end{bmatrix} = \begin{bmatrix} X_t^{cc}(k,l) \\ X_t^{cs}(k,l) \\ X_t^{sc}(k,l) \\ X_t^{ss}(k,l) \end{bmatrix}$$

where

$$f(k,l) = CS(k,l); \text{ for } k,l \in \{1,\ldots,N-1\},$$
$$g(k,l) = SC(k,l); \text{ for } k,l \in \{1,\ldots,N-1\}.$$

At the block boundaries in the transform domain, the pseudo-phase functions can be found as in the 1-D case. These pseudo-phase functions are then passed through 2D-DCT-II decoders (inverse DCSTII and DSCTII coders) to generate two δ functions, $DCS(\cdot,\cdot)$ and $DSC(\cdot,\cdot)$. The estimated displacement, $(\hat{m}_1, \hat{n}_1)$, can then be found by locating the peaks of DCS and DSC over $\{0,\ldots,N-1\}^2$ or an index range of interest, usually, $\{0,\ldots,N/2\}^2$ for slow motion. The signs of the peaks will determine the directions of movements according to Table. 1. Typical DCS and DSC are depicted in Fig. 2 where DXT-ME is performed on the images of a moving object of arbitrary shape corrupted by additive white Gaussian noise at SNR = 10.

The computational complexity of the modules of computing $f(\cdot,\cdot)$, $g(\cdot,\cdot)$ and determining the displacement is $O(N^2)$ for a block of size N^2. Thus, with DCT

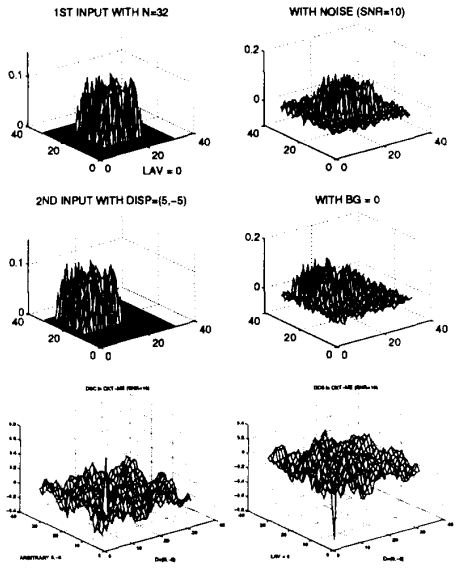

Figure 2: DXT-ME performed on the images of an object moving in the direction (5, -5) with additive white Gaussian noise at SNR = 10 dB

codecs having the complexity O_{dct}, the overall complexity of DXT-ME is $O(N^2) + O_{dct}$. If we adopt the parallel and fully-pipelined 2D DCT lattice structure with a complexity of $4N$ [4], then the complexity of DXT-ME will remain as $O(N^2)$, much lower compared to $O(N^4)$, the complexity of BMA-ME.

4. PREPROCESSING AND SIMPLIFIED EXTENDED DXT-ME

For more complex video sequences, in which moving objects are moving across the border of blocks in non-darken background, preprocessing must be employed to avoid any violation of the assumption for DXT-ME before feeding the images into the DXT-ME motion estimator; otherwise, the performance will be jeopardized. Preprocessing does not affect the accuracy of motion estimation if the preprocessing function does not distort or destroy the information of motion in the original sequence because DXT-ME estimates the motion of an object regardless of the shape of the object as long as the block contains significant energy level of signal features of the object. In order to maintain the advantage of this DXT-ME as a low-complexity motion estimator, only simple preprocessing is used on such video sequences. In the simulation below, edge detection and frame differentiation are used for the sake of simplicity and capability of extraction of motion information. The computational complexity of this step is only $O(N^2)$.

Sign of DSC Peak	Sign of DCS Peak	Peak Index
+	+	(m_1, n_1)
+	−	$(m_1, -(n_1+1))$
−	+	$(-(m_1-1), n_1)$
−	−	$(-(m_1+1), -(n_1+1))$

Table 1: Determination of Direction of Movement (m_1, n_1) from the Signs of DSC and DCS

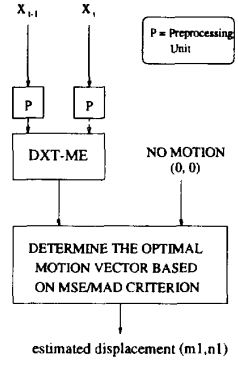

Figure 3: Block Diagram of Simplified Extended DXT-ME

The simplicity of the preprocessing function and the complication of most video sequences make it possible that spurious DCS or DSC peaks might adversely affect the accuracy of the motion estimates by DXT-ME. To lessen this effect, a simplified extended DXT-ME structure, as depicted in Fig. 3, is devised to choose, among the DXT-ME estimate and no motion, the optimal displacement vector according to the MSE criterion.

5. SIMULATION RESULTS

The first video sequence, SCAR (acronym for small car), contains only two frames of images. The first frame of SCAR (SCAR_1) is manually shifted to produce the second frame (SCAR_2) with a known displacement and additive Gaussian noise is added to attain a desired signal-to-ratio (SNR) level. The object in this sequence moves within the boundary of the frame under the completely darken background. As can be seen in Fig. 4, DXT-ME is performed on the whole image of block size 64 × 64 and estimates the motion correctly at SNR level even down to 0 dB, whereas BMA-ME produces some wrong motion estimates for boundary blocks and blocks of low signal energy. The values of MAD also indicate better overall performance of DXT-ME over BMA-ME for this sequence. Here MAD means Minimum Absolute Difference defined as follows:

$$MAD = \frac{\sum_{m,n} |\hat{x}(m,n) - x(m,n)|}{N^2}.$$

Furthermore, DXT-ME can be performed on the whole frame while BME-ME needs division of the frame into

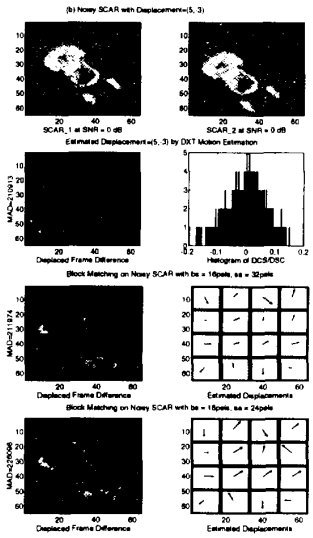

Figure 4: Comparison of DXT Motion Estimation Scheme of Size 64 by 64 pels with Block Matching Method of Block Size (bs = 16 pels) but Different Search Areas (sa = 32 or 24 pels) on Noisy SCAR with SNR = 0 dB.

sub-blocks due to the requirement of larger search areas than reference blocks. This is one of the reasons that BMA-ME does not work so well as DXT-ME because smaller block size makes BMA-ME more susceptible to noise but operation of DXT-ME on the whole frame instead of on smaller blocks lends itself to better noise immunity.

The second video 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. Simple preprocessing is done on this sequence: edge detection and frame differentiation. Simulation was performed on 19 frames of this sequence with DXT-ME of the block size = 16, 32, 64, and BMA-ME of the block size = 16, 32 and the search area being twice as large as the block size. Absolute differences of two consecutive frames with motion compensation (DIF in all the figures of results) are also computed for comparison. The results of the performance of both schemes are shown in Fig. 6, where MSE is computed as below:

$$MSE = \frac{\sum_{m,n}[\hat{x}(m,n) - x(m,n)]^2}{N^2},$$

Here $x(m,n)$ is the pixel value of the original image at position (m,n) and $\hat{x}(m,n)$ is the pixel value of the reconstructed image based upon the displacement field estimated by either method. In the simulation, BMA-ME scheme is the full block matching method to use the MSE criterion. In other words, it minimizes the MSE function over the whole search area:

$$\hat{d} = (\hat{u}, \hat{v}) = arg \min_{u,v} \frac{\sum_{m,n}[x_2(m,n) - x_1(m-u, n-v)]^2}{N^2}.$$

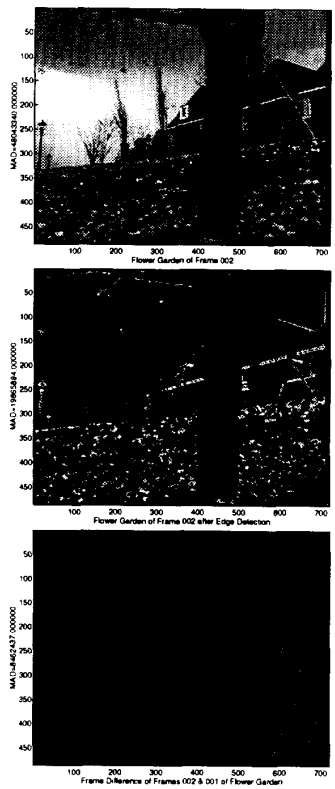

Figure 5: Sequence "Flower Garden" (FG)

As can be seen in Fig. 5, the edge detected 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 detected 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. Meanwhile, due to the constant movement of the camera, the energy of the moving objects in the frame differentiated sequence of 'Flower Garden' is strong enough for DXT-ME to estimate the motion directly on this frame differentiated sequence.

Observable in Fig. 6, increasing block size for BMA-ME hampers its performance with increasing MSE values whereas increasing block size of DXT-ME gives us better performance with smaller MSE values. The reason is that a block of larger size for DXT-ME contains more features of objects, which enables DXT-ME to find a better estimate, and also a block of larger size means a larger search area because the size of a search area is the same as the block size for DXT-ME. As a matter of fact, BMA-ME is supposed to perform better than DXT-ME if both methods use the same block size because BMA-ME requires a larger search area and thus BMA-ME has more information available before processing than DXT-

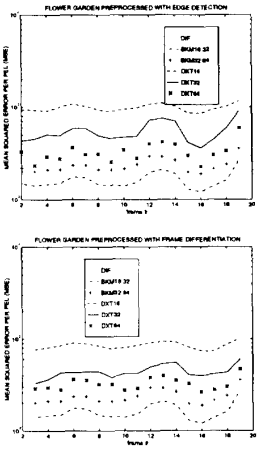

Figure 6: Comparison of DXT Motion Estimation Scheme with Full Block Matching Method on Sequence "Flower Garden"

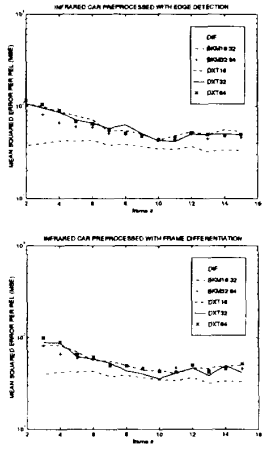

Figure 8: Comparison of Simplified Extended DXT-ME with BMA-ME on Sequence "Infrared Car"

ME. Therefore, it is not fair to compare BMA-ME with DXT-ME for the same block size. Instead, it is more reasonable to compare BMA-ME with DXT-ME of the block size equal to the size of the search area of BMA-ME. As shown in Fig. 6, the MSE values for DXT-ME of block size 64 preprocessed by either edge detection or frame differentiation are comparable with those for BMA-ME of block size 32 and search size 64.

Another simulation was done on the sequence, "Infrared Car" which has only one observable moving object, the car moving along the curved road towards the camera fixed on the ground in Fig. 7. The simplified extended DXT-ME is applied to this sequence. As shown in Fig. 8, DXT-ME for the block sizes 64, 32 or even 16 can perform as well as BMA-ME of block size 32 and search size 64.

6. SUMMARY

DXT-ME computes the DCT pseudo-phase of images and employs the orthogonal principles of DCT/DST kernels to estimate motions. In this way, it can be incorporated into codecs of various image compression protocols like MPEG, CCITT H261, etc. and enables us to utilizes the advancement of DCT codecs which is under extensive research. Above all, it requires much less computational complexity $O(N^2)$ as compared to $O(N^4)$ for BMA-ME and has very good noise immunity. In situations of violating the assumption, simple preprocessing is needed before DXT-ME. Even though preprocessing usually decreases SNR, the performance is still reasonable.

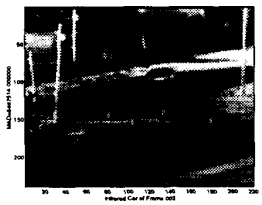

Figure 7: Sequence "Infrared Car" (CAR)

7. REFERENCES

[1] D. J. Le Gall, "MPEG: A Video Compression Standard For Multimedia Applications," Communications of the ACM, Vol34, No4, April 1991.

[2] M. Liou, "Overview of the px64 kbit/s Video Coding Standard," Communications of the ACM, Vol34, No4, April 1991.

[3] P. Yip and K. R. Rao, "On the Shift Property of DCT's and DST's," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-35, No. 3, March 1987.

[4] C. T. Chiu and K. J. Liu, "Real-Time Parallel and Fully-Pipelined Two-Dimensional DCT Lattice Structures with Application to HDTV Systems," IEEE Transactions on Circuits and Systems for Video Technology, pp.25-37, Vol.2, No.1, March 1992.

[5] A. Zakhor and F. Lari, "Edge-Based 3-D Camera Motion Estimation with Application to Video Coding," IEEE Transactions on Image Processing, pp.481-498, Vol.2, No.4, October 1993.

[6] A. K. Jain, "Fundamental of Digital Image Processing," Prentice-Hall, 1989.