DCT-BASED PHASE CORRELATION MOTION ESTIMATION

Min Li, Mainak Biswas, Sanjeev Kumar and Truong Nguyen

UCSD, ECE Dept., La Jolla CA 92093

ABSTRACT

A DCT-based phase correlation motion estimation algorithm is proposed in this paper. By combining four real transforms, a new complex linear phase transform is obtained and is used in phase correlation motion estimation. The application in video compression is discussed in details. The simulation results show that the proposed algorithm is robust and efficient.

1. INTRODUCTION

Video compression techniques that remove temporal and spatial redundancy in a video sequence, are used in video transmission and storage applications [1]. A fast and accurate motion estimation algorithm is essential for an efficient video codec. The traditional block-matching motion estimation algorithm is a spatial domain method and employs the Mean Absolute Error (MAE) or the Mean Square Error (MSE) as criteria to determine the motion vector. However, the MAE or MSE criteria only measures the maximal or average difference between each pair of pixels, which does not resemble the visual system's behaviour. According to the Gestalt theory [2], the visual system tends to assemble some components of a picture and perceive them together. Besides, in order to achieve a good motion estimation, the calculation task of the block-matching method is exhaustive.

The phase-based motion estimation is an alternate approach for motion estimation. In the phase-based motion estimation method, the block is processed as a whole, which resembles the visual system's behaviour. In addition, there is no 'search' among blocks in phase-based motion estimation algorithm, rather, the phase correlation is carried out only between two corresponding blocks before determining the motion vector. Thus, the calculation task is relatively small and the phase-based motion estimation is an efficient and fast motion estimation algorithm. The Fourier Transform (FT)-based motion estimation [3] and the Discrete Cosine Transform (DCT)-based motion estimation [4] are two main phase-based motion estimation approaches.

This work is supported by Office of Naval Research. The emails of the authors are: min_li, nguyent@ece.ucsd.edu, mbiswas, sakumar@ucsd.edu

Shift invariant property of the FFT is the operational principle of the FFT-based phase correlation motion estimation algorithm. The spatial domain shift causes only the phase changes in frequency domain and the shift information can be detected from the phase changes in frequency domain. The FFT-based phase correlation motion estimation method is efficient in terms of motion estimation. However, it is not so efficient for video compression application due to the fact that both DCTII and FFT have to be performed upon each image block.

DCT-based motion estimation receives more attention for video compression applications, since DCTII is employed in the current most popular video compression standards, MPEGs. Taking advantage of the already available DCTII coefficients, some calculation resources for motion estimation can be saved if the motion estimation algorithm is DCT-based. Consequently, small latency can be expected if the DCT-based motion estimation algorithm is applied to the video codec.

Chen, Koc and Liu in [4] proposed a DCT-based motion estimation algorithm which works for 1D signals. In the 2D case, the algorithm fails. According to the simulation results presented in [4], the algorithm might work with respect to 2D signals if the input 2D signals are central symmetric or be edge-extracted. However, the practical image signals can not be assumed to be central symmetric. On the other hand, some pre-processing has to be done in order to achieve the edge-extracted images. The pre-processing process has two main problems. First, the extra pre-processing process definitely increases the computation complexity. Second, the low frequency information in an image is thrown away during the edge extraction process. However, there is no reason to assume that motion in a video sequence only occurs on edge areas as a plain block can shift to another position.

In summary, the proposed DCT-based motion estimation algorithm in [4] is inaccurate and inefficient. In this paper, we present a robust DCT-based phase correlation motion estimation algorithm. Compared to the algorithm proposed in [4], our proposed algorithm works in terms of original image signals, without pre-processing. In addition, only 4 of the 8 transforms, which are required to be performed on each frame in [4], are used in our proposed algorithm. The calculation complexity decreases significantly.

The organization of this paper is as follows. The DCT-

0-7803-8554-3/04/$20.00 ©2004 IEEE.

based phase correlation motion estimation algorithm is presented in Section II. The DCTII domain motion compensation is discussed in Section III. Various implementation aspects are discussed in Section IV, which include block windowing and filtering and motion vector determination. Simulation results and discussions are presented in Section V.

2. DCT-BASED PHASE CORRELATION MOTION ESTIMATION

As discussed in Section I, although FFT-based phase correlation motion estimation method is efficient in performing motion estimation, it is not so efficient for video compression application. In this section, we develop a DCT-based phase correlation motion estimation algorithm such that both motion estimation and video compression are done based on the DCTII or some DCTII-related transforms as

$$X_{cc}(K_1,K_2) = C_1C_2\sum_{m,n=0}^{N-1} x(m,n)\cos[\frac{\pi K_1}{N}(m+.5)]\cos[\frac{\pi K_2}{N}(n+.5)]$$

$$X_{cs}(K_1,K_2) = C_1C_2\sum_{m,n=0}^{N-1} x(m,n)\cos[\frac{\pi K_1}{N}(m+.5)]\sin[\frac{\pi K_2}{N}(n+.5)]$$

$$X_{sc}(K_1,K_2) = C_1C_2\sum_{m,n=0}^{N-1} x(m,n)\sin[\frac{\pi K_1}{N}(m+.5)]\cos[\frac{\pi K_2}{N}(n+.5)]$$

$$X_{ss}(K_1,K_2) = C_1C_2\sum_{m,n=0}^{N-1} x(m,n)\sin[\frac{\pi K_1}{N}(m+.5)]\sin[\frac{\pi K_2}{N}(n+.5)],$$

where C_1 and C_2 are constants and defined as

$$C_i = \begin{cases} \sqrt{\frac{2}{N}}, & K_i \in \{1,\cdots,N-1\} \\ \sqrt{\frac{1}{N}}, & K_i = 0 \end{cases}, i = 1,2. \quad (1)$$

Among the four transforms defined above, the first transform is exactly the DCTII, which is used in MPEG standards. The other three transforms can be implemented using the similar hardware structure as DCTII. If the four real transforms are combined as $\{X_{cc}(K_1,K_2) - X_{ss}(K_1,K_2)\} - j\{X_{cs}(K_1,K_2) + X_{sc}(K_1,K_2)\}$, a complex transform is obtained and expressed as

$$X(K_1,K_2) = C_1C_2\sum_{m,n=0}^{N-1} x(m,n)e^{-j\frac{\pi K_1}{N}(m+.5)}e^{-j\frac{\pi K_2}{N}(n+.5)}. \quad (2)$$

The complex transform in eq.(2), named DCT-based complex transform, is a shift invariant transform, which can be used to perform phase correlation motion estimation. The operational principle is as follows. Let $X_p(K_1,K_2)$ and $X_c(K_1,K_2)$ denote the DCT-based complex transforms of the 2D signal $x(m,n)$ and $x(m-m_0,n-n_0)$ respectively. Both $x(m,n)$ and $x(m-m_0,n-n_0)$ are of size $N\times N$ and one is the other's shifted version, then $X_c(K_1,K_2)$ can be expressed as

$$X_c(K_1,K_2) = X_p(K_1,K_2)e^{-j\frac{\pi K_1}{N}m_0}e^{-j\frac{\pi K_2}{N}n_0}. \quad (3)$$

The normalized phase correlation function of $X_c(K_1,K_2)$ and $X_p(K_1,K_2)$ can be expressed as

$$\frac{X_c(K_1,K_2)X_p^*(K_1,K_2)}{\|X_c(K_1,K_2)X_p^*(K_1,K_2)\|} = e^{-j\frac{\pi K_1}{N}m_0}e^{-j\frac{\pi K_2}{N}n_0}. \quad (4)$$

Applying the inverse DFT upon the normalized phase correlation function of $X_c(K_1,K_2)$ and $X_p(K_1,K_2)$, the result is

$$\mathcal{IDFT}\{e^{-j\frac{\pi K_1}{N}m_0}e^{-j\frac{\pi K_2}{N}n_0}\} = \delta(m+\frac{m_0}{2},n+\frac{n_0}{2}). \quad (5)$$

The motion vector information (m_0,n_0) can be determined by locating the impulse peak on the phase correlation plane.

The procedures above are summarized using the block flow diagram shown in Fig.1. A (24×24) 2D random signal $x(m,n)$ and its shifted version $x(m-m_0,n-n_0)$ are shown in Fig.2. The corresponding phase correlation plane is shown in Fig.2(c). The impulse peak position on the phase correlation plane indicates the motion vector (m_0,n_0).

The complex transform defined in eq.(2) has shift invariant property, so it can be used to perform phase correlation motion estimation. It is very efficient for video compression application since it is DCT-based. If the DCTII, among the four real transforms, is implemented efficiently using a hardware structure, the other three real transforms can be implemented as efficiently as DCTII using similar structures. So the implementation of the complex transform defined in eq.(2) is very efficient.

3. MOTION COMPENSATION IN TRANSFORM DOMAIN

After motion estimation, the traditional way to obtain the DCTII coefficients of the error frame is to do motion compensation in the spatial domain and then to perform DCTII on the predicted error frame. It's not efficient to go back to the spatial domain to perform motion compensation. In the proposed method, the motion estimation is performed in transform domain. It would be relatively efficient if the motion compensation is done in transform domain such that the DCTII coefficients of the error frame are obtained directly.

If the complex transform defined in eq.(2) is implemented using four real transforms, the DCTII coefficients can be stored separately. Thus, after motion estimation, the DCTII coefficients of each $M\times M(M=32)$ block are available.

Chang in [5] suggested one way to extract the DCTII coefficients of a certain $N\times N$ sub-block $\mathbf{B}_{N\times N}$ from the DCTII coefficients of a $M\times M$ block, $\mathbf{B}_{M\times M}, M>N$. It is a fact that, if only $\mathbf{B}_{N\times N}$ is a sub-block of $B_{M\times M}$, there exist matrices $\mathbf{R}$ and $\mathbf{C}$ such that the sub-block $\mathbf{B}_{N\times N}$ can be extracted from the block $\mathbf{B}_{M\times M}$ as

$$\mathbf{B}_{N\times N} = \mathbf{R}\mathbf{B}_{M\times M}\mathbf{C}. \quad (6)$$

Applying DCTII to block $\mathbf{B}_{N\times N}$ in eq.(6) and also applying the orthogonal property of the DCTII, we get

$$\mathcal{DCTII}(\mathbf{B}_{N\times N}) = \mathcal{DCTII}(\mathbf{R})\mathcal{DCTII}(\mathbf{B}_{M\times M})\mathcal{DCTII}(\mathbf{C}), \quad (7)$$

where $\mathcal{DCTII}(\mathbf{R})$ and $\mathcal{DCTII}(\mathbf{C})$ can be pre-calculated and stored, such that the calculation in eq.(7) is efficient.

The DCTII coefficients of each 16×16(the motion compensation unit size) block can be extracted from the available DCTII coefficients of each 32×32 block according to eq.(6) and eq.(7). In addition, the prediction block can be thought as being composed of four sub-blocks that are sub-blocks of the motion compensation unit blocks $\mathbf{B}_1,\mathbf{B}_2,\mathbf{B}_3$ and $\mathbf{B}_4$. Each sub-block can be extracted from the corresponding motion compensation unit block with a pre-matrix $\mathbf{R}_i$ and

a post-matrix $\mathbf{C}_i$, and, consequently, the DCTII coefficients of each prediction block $\hat{\mathbf{B}}$ can be expressed as

$$\mathcal{DCTII}(\hat{\mathbf{B}}) = \sum_{i=1}^{4} \mathcal{DCTII}(\mathbf{R}_i)\mathcal{DCTII}(\mathbf{B}_i)\mathcal{DCTII}(\mathbf{C}_i), \quad (8)$$

where the DCTII coefficients of matrices $\mathbf{R}_i$ and $\mathbf{C}_i, i = 1, \cdots, 4$ are pre-calculated and stored. Thus, the DCTII coefficients of the prediction block $\hat{\mathbf{B}}$ can be obtained directly from the DCTII coefficients of the motion compensation unit blocks $\mathbf{B}_i, i = 1, \cdots, 4$. The DCTII coefficients of the error frame can be obtained by subtracting the DCTII coefficients of the prediction frame from that of the P frame. In practice, instead of the exact $\mathcal{DCTII}(\mathbf{B}_i)$, the quantized $\mathcal{DCTII}(\mathbf{B}_i)$ is used, of which most entries are zeros, so the calculation in eq.(8) can be very efficient. Some methods to compute eq.(8) efficiently are suggested in [6][7].

In summary, motion compensation can be performed in DCTII domain. Together with the DCT-based phase correlation motion estimation, a complete transform domain codec is obtained.

4. IMPLEMENTATION DETAILS

After dividing each frame into $M \times M$ blocks, windowing (or filtering) has to be performed on each block in order to obtain accurate motion vectors. In addition, the phase correlation plane may have more than one peaks with comparable levels in practice. This is because there may be more than one dominant motion on a single block. Some steps have to be followed to pick out the exact motion vector.

4.1. Windowing and filtering

The transform defined in eq.(2) relates DFT in the following way. The transform in eq.(2) can be rewritten as

$$X(K_1, K_2) = W_c(K_1, K_2)F_{ft}\{x(m, n)\}, \quad (9)$$

where $W_c(K_1, K_2) = C_1C_2e^{-j\frac{\pi K_1}{2N}}e^{-j\frac{\pi K_2}{2N}}$ is a 2D complex window applied in the transform domain, and $F_{ft}\{x(m,n)\} = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x(m,n)e^{-j\frac{\pi K_1}{N}m}e^{-j\frac{\pi K_2}{N}n}$ is actually an FFT2, half of the calculation points are the same as the regular FFT in 1D case.

It's observed that the complex window $W_c(K_1, K_2)$ does not affect the phase difference term $e^{-j\frac{\pi K_1}{N}m_0}e^{-j\frac{\pi K_2}{N}n_0}$. Thus, if only the phase term F_{ft} be estimated correctly, the phase difference $e^{-j\frac{\pi K_1}{N}m_0}e^{-j\frac{\pi K_2}{N}n_0}$ can be extracted correctly.

Windowing and filtering play important roles in the accurate numerical estimation of F_{ft}. According to the definition of DFT, the discrete spectrum $X(K)$ of a length limited discrete signal $x(n)$ is the spectrum of the signal $\tilde{x}(n)$, which is periodic and equal to $x(n)$ over one period. The signal $\tilde{x}(n)$ can have frequency components that are not contained in the signal $x(n)$. As shown in Fig.3(c), a length-24 signal is extracted from a length-47 signal shown in Fig.3(a), the spectrum of the extracted signal shown in Fig.3(d) has much more high frequency components compared to the spectrum of the original signal, as shown in Fig.3(b). Windows or filters are applied to the extracted signal such that no or fewer frequency components are created when it is extended periodically. The windowed signal is shown in Fig.3(e). Note that the high frequency components in its spectrum shown in Fig.3(f) has much lower level compared to the spectrum shown in Fig.3(d). The high frequency components are constrained much more when a filter is applied to the extracted signal, which can be observed from the spectrum of the filtered signal, as shown in Fig.3(h). Filtering performs better in terms of reducing the extra frequency. However, filtering has higher complexity than windowing since convolution is used.

4.2. Determine the motion vector on the phase correlation plane

The following steps are used to determine the motion vectors. First, locate the top 5 possible peaks and save their position coordinates and levels. Second, sort the 5 pairs of coordinates according to the peak levels and then check

- if the first peak level is significantly high. If it is, the motion vector is determined using the first peak position coordinates.
- if the coordinates of any two adjacent peaks are neighbors or partially the same, and, at the same time, if
  1. their peak level difference is less than a certain percentage of either of the two peak levels or
  2. it is not the maximal peak level difference or
  3. the neighbors of the pair of peaks have a much lower or higher level than either of the two.

  If they are, the motion vector is calculated by combining their position coordinates. The necessity to combine the position coordinates is that the discrete transform is numerical estimation of samples on the actual curve. The impulse peak can occur between the samples.
- If no motion vectors are determined, choose the one that minimizes the MSE of the error block.

5. SIMULATION RESULTS AND DISCUSSIONS

Some simulation results are presented in this section to show the performance of the proposed algorithm. Two frames from the light sequence are shown in Fig.4(a) and (b). Using the proposed method, the resulting motion vector field is shown in Fig.4(c) and the motion vector field generated by the full search algorithm is shown in Fig.4(d). Note that the

proposed method yields a much cleaner motion vector field and requires fewer bits to transfer the motion vector field.

The MSE of the resulting error frames generated by the proposed method and that of the error frames generated by the full search method are shown in Fig.5. In the simulation, about 5% of the blocks are treated as I-blocks.

6. CONCLUSIONS AND FUTURE WORK

We presented a DCT-based phase correlation motion estimation algorithm in this paper which is robust and efficient. The calculation complexity of the proposed algorithm is relatively low compared to previous works. Future works include incorporating the proposed motion estimation into MPEG standards and compare its performance to the MPEG standards.

7. REFERENCES

[1] P. Symes, *Video compression demystified*, McGraw-Hill, 2001.

[2] I. Gordon, *Theories of Visual Perception*, John Wiley and Son, 1997.

[3] M. Biswas and T. Nguyen, "A novel motion estimation algorithm using phase plane correlation for frame rate conversion," *Asilomar Conf.*, vol. 1, pp. 492–496, Nov. 2002.

[4] J. Chen, U.-V. Koc, and K.J.R. Liu, *Design of digital video coding systems: a complete compressed domain approach*, Marcel Dekker, New York, 2002.

[5] S. Chang and D.G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video," *IEEE Journal on selected areas in communications*, vol. 13, no. 1, pp. 1–11, Jan. 1995.

[6] N. Merhav and V. Bhaskaran, "A fast algorithm for DCT-domain inverse motion compensation," *Proc. ICASSP*, vol. IV, pp. 2307–2310, May 1996.

[7] J. Song and B.-L. Yeo, "A fast DCT-domain inverse motion compensation algorithm based on shared information in a microblock," *Asilomar Conf.*, vol. 1, pp. 843–847, Nov. 1998.

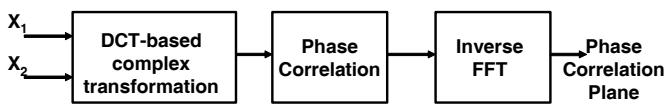

Fig. 1. Steps in the proposed algorithm.

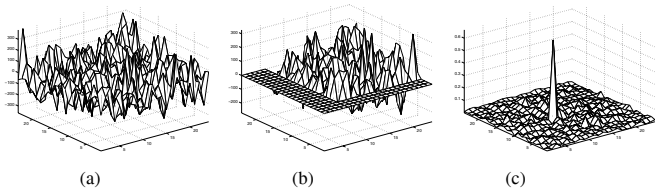

Fig. 2. (a) A 2D (24×24) random signal $x(m, n)$; (b) the shifted 2D signal $x(m-4, n-6)$; (c) the resulting phase correlation plane and the peak indicates the motion vector.

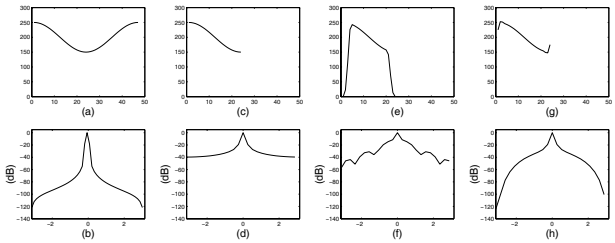

Fig. 3. (a) A length-47 1D signal, $f(t) = 200 + 50\cos(t)$; (b) magnitude spectrum of signal shown in (a); (c) first 24 samples of the signal in (a); (d) magnitude spectrum of the signal shown in (c); (e) windowed signal; (f) magnitude spectrum of the signal shown in (e); (g) filtered signal; (h) magnitude spectrum of the signal shown in (g).

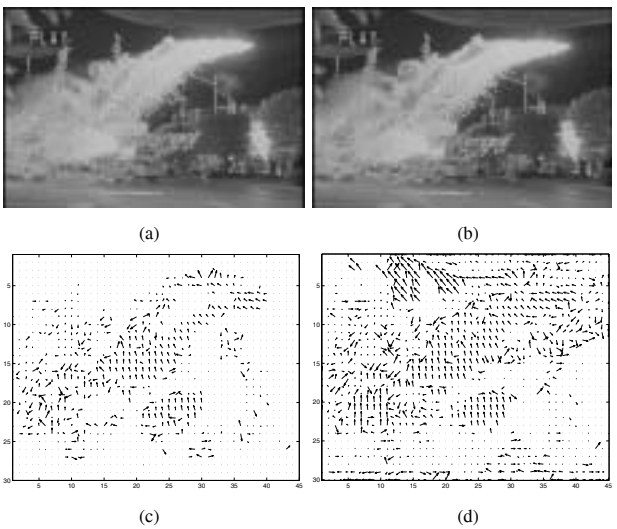

Fig. 4. (a) I frame; (b) P frame; (c) motion vector field generated by the proposed method; (d) motion vector field generated by the full search method.

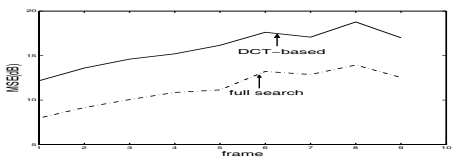

Fig. 5. MSE of the predicted error frames.