

Low-Complexity Sub-Pixel Motion Estimation Utilizing Shifting Matrix in Transform Domain

Chul Ryu*, Jae-Young Shin* and Eun-Chan Park[†]

Abstract – Motion estimation (ME) algorithms supporting quarter-pixel accuracy have been recently introduced to retain detailed motion information for high quality of video in the state-of-the-art video compression standard of H.264/AVC. Conventional sub-pixel ME algorithms in the spatial domain are faced with a common problem of computational complexity because of embedded interpolation schemes. This paper proposes a low-complexity sub-pixel motion estimation algorithm in the transform domain utilizing shifting matrix. Simulations are performed to compare the performances of spatial-domain ME algorithms and transform-domain ME algorithms in terms of peak signal-to-noise ratio (PSNR) and the number of bits per frame. Simulation results confirm that the transform-domain approach not only improves the video quality and the compression efficiency, but also remarkably alleviates the computational complexity, compared to the spatial-domain approach.

Keywords: Motion estimation, Transform domain, Recursive equation, Quarter-pixel accuracy, Computational complexity

1. Introduction

The recent advances in mobile communication technologies facilitate diverse multimedia services in a wireless environment, and thus, the demands of high-quality video services in mobile devices are continuously increasing. In order to satisfy these demands, video compression is an imperative technique because high-quality video requires a great amount of information to be transmitted and received over a bandwidth-limited wireless channel. ITU-T and ISO jointly developed and proposed H.264/AVC aiming to provide high-quality video at substantially lower bit rates than the previous standards (e.g., MPEG-2 and H.263) without increasing the design complexity substantially. Therefore, H.264/AVC became one of the most prevailing video codecs for recording, compression, and distribution of video contents, e.g., Internet video streaming and HDTV broadcasting, and it is widely embedded in various consumer electronics such as digital cameras/camcorders, digital TV set-top boxes, mobile phones, and CCTVs, as well desktops and laptops.

The essence of video compression is to reduce spatial and temporal redundancies, and a motion estimation (ME) algorithm is the key part of video compression to remove temporal redundancy among consecutive frames in the video sequence. In literature, ME algorithms have been continuously and widely studied by many researchers to improve the compression efficiency and/or to lower the

computational complexity, because the computational complexity of ME takes up about 60%-80% of total computational complexity of video compression [1]. For the purpose of motion estimation, a block matching algorithm is usually used to estimate a motion vector by finding a location that gives the minimum matching error within the search area in the previous frame. A video encoder transfers the motion vectors found by the ME algorithm and the errors between the current and the best-matched blocks using entropy coding. More precise motion vectors can reduce the temporal redundancy effectively, because they lead to less amount of error between the current and the target block found by motion estimation.

Motion vector does not always occur in integer-pixels, it may also occur in sub-pixels due to changes of brightness among frames, nonlinear motion, or objects being appeared or disappeared in the boundary of pixel. Therefore, it cannot be expected to obtain precise motion information with a ME algorithm supporting integer-pixel accuracy, and it is necessary to adopt a sub-pixel ME algorithm for achieving more accurate motion vectors. For this reason, MPEG-2 and H.263 utilize half-pixel accuracy ME and H.264 further expands it to the level of quarter-pixel accuracy to achieve better video quality than the existing video coding standards.

Conventional ME algorithms are usually performed in the spatial domain. However, the sub-pixel ME algorithms in the spatial domain result in a serious increment of computational complexity of video encoders due to embedded interpolation algorithms. To overcome this drawback, this paper proposes a low-complexity ME algorithm based on a transform-domain approach. By using shifting matrix in the transform domain and predefined

[†] Corresponding Author: Dept. of Information and Communication Engineering, Dongguk University, Seoul, Korea. (ecpark@dongguk.edu)

* Dept. of Information and Communication Engineering, Dongguk University, Seoul, Korea. (cryu@dongguk.edu, jyshin702@gmail.com)

Received: September 24, 2015; Accepted: February 3, 2016

coefficient of interpolation, the proposed ME algorithm significantly reduces the computational complexity without impairing the video quality. The simulation results confirm that the proposed transform-domain approach outperforms the conventional spatial-domain approach in terms of the peak signal-to-noise ratio (PSNR) and the number of bits per frame, which are effective measures for the video quality, compression efficiency, and computational complexity.

The paper is organized as follows: Section 2 describes the ME algorithm in the transform domain. In Section 3, a shifting matrix algorithm is proposed for quarter-pixel accuracy ME. Section 4 presents simulation results comparing the performance of spatial-domain ME algorithms with that of transform-domain ME algorithms. Section 5 concludes and addresses the utilization of the proposed algorithm.

2. Motion Estimation in Transform Domain

Since video encoding and decoding are performed in a unit of block, block-based ME algorithms are preferred. Block-based ME algorithms can be classified as spatial-domain algorithms and transform-domain algorithms. The common spatial-domain ME algorithms are block matching algorithm and gradient-based algorithm, while the typical transform-domain ME algorithms are discrete cosine transform (DCT) and wavelet-based algorithm [2-6].

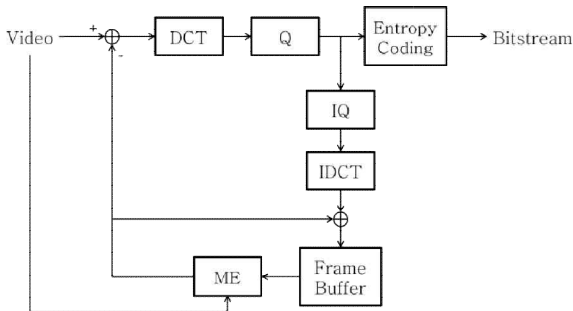


Fig. 1. Motion estimation and compensation of video encoder in the spatial domain

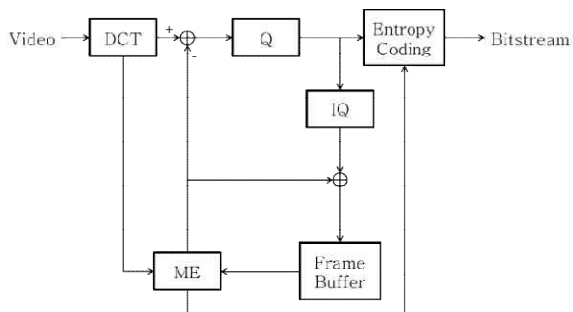


Fig. 2. Motion estimation and compensation of video encoder in the transform domain

Among the transform-domain algorithms, the wavelet-based algorithm spends less time in searching and matching; however, it incurs a high complexity in estimating motion vectors and increases the bit rate to transmit motion vectors in all the sub-bands. On the other hand, DCT-based algorithm is widely used in various video compression standards, and thus, it maintains a high compatibility to many existing standards.

DCT transforms the spatial-domain data into the transform-domain coefficients. One of its desirable features is that the energy of transformed coefficients is concentrated on the low frequency bands, which can be a basis for high efficiency of video compression. In general, the spatial-domain motion estimation requires the process of inverse DCT (IDCT) in order to utilize the spatial-domain information of the restored frame which is used as a reference frame in the estimation process, as shown in Fig. 1. Here, Q and IQ denote quantization and inverse quantization, respectively. On the other hand, IDCT can be omitted in the transform-domain ME since the motion estimation can be performed using coefficients transformed by DCT, as illustrated in Fig. 2 [7]. One of the key advantages of transform-domain ME is the low computational complexity, which is originated from the simple encoder structure without IDCT.

The shifting matrix algorithm proposed by Plompen showed that a prediction block f_{pred} can be represented with the horizontal and vertical conversion of neighboring blocks [8]. Fig. 3 shows an example that four neighboring blocks in the previous frame, $f_1, f_2, f_3,$ and f_4 , are used to estimate the f_{pred} which is the best-matched block to the current block f_{curr} .

The prediction block f_{pred} can be expressed with a vertical shifting matrix h_i and a horizontal shifting matrix v_i , i.e.,

$$f_{pred} = \sum_{i=0}^3 v_i f_i h_i \quad (1)$$

where, h_i and v_i in Eq. (1) are defined as follows.

$$h_0 = D_{4-\Delta x}^t, \quad h_1 = D_{\Delta x}, \quad (2)$$

$$h_2 = D_{4-\Delta x}, \quad h_3 = D_{\Delta x}$$

$$v_0 = D_{4-\Delta y}, \quad v_1 = D_{4-\Delta y}, \quad (3)$$

$$v_2 = D_{\Delta y}, \quad v_3 = D_{\Delta y}$$

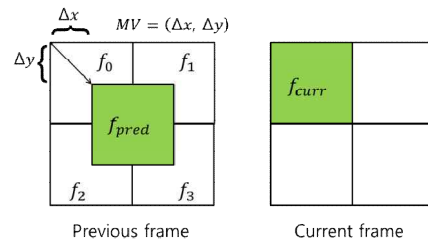


Fig. 3. Estimation of current block from the best- matched block composed of neighboring blocks in the previous frame

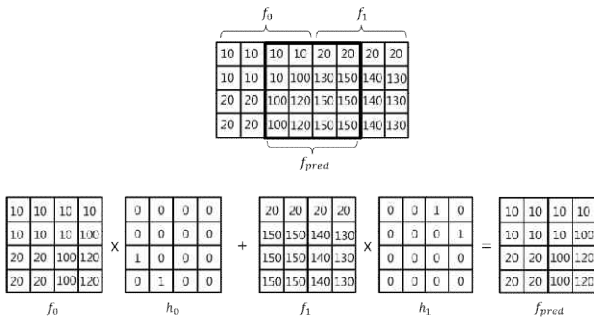


Fig. 4. Example of estimation process of f_{pred} with $\Delta x = 2, \Delta y = 0$

Here, the displacement matrix D_n is defined as

$$D_n = \begin{bmatrix} 0 & I_n \\ 0 & 0 \end{bmatrix}, \quad (4)$$

where, I_n is a $n \times n$ unit matrix. For example, Fig. 4 demonstrates a process of predicting f_{pred} with $\Delta x = 2, \Delta y = 0$, and Eq. (5) and (6) represent h_i used in this example.

$$h_1 = h_3 = D_2^t = \begin{bmatrix} 0 & 0 \\ I_2 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (5)$$

$$h_0 = h_2 = D_2 = \begin{bmatrix} 0 & I_2 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (6)$$

Applying orthogonality and separability of DCT, F_{pred} , DCT coefficient of the prediction block, can be expressed from Eq. (1) as follows:

$$F_{pred} = DCT(f_{pred}) = \sum_{i=0}^3 DCT(v_i) DCT(f_i) DCT(h_i) \quad (7)$$

$$= \sum_{i=0}^3 (T v_i T^t) (T f_i T^t) (T h_i T^t) = \sum_{i=0}^3 V_i F_i H_i$$

Note that F_{pred} can be obtained by carrying out eight operations of matrix multiplication and three operations of matrix subtraction. However, the amount of computation can be greatly reduced by utilizing the characteristic of energy concentration of DCT. Accordingly, Eq. (7) can be derived as follows, in order to relieve the computational complexity by taking the advantage of DCT transformed block F_i , i.e.,

$$F_{pred} = \sum_{i=0}^3 V_i F_i H_i = \sum_{i=0}^3 \sum_{m=0}^3 \sum_{n=0}^3 F_i(m, n) V_i(m) H_i(n)$$

$$= \sum_{m=0}^3 \sum_{n=0}^3 \sum_{i=0}^3 F_i(m, n) V_i(m) H_i(n) \quad (8)$$

All $F_i(m, n)$ with zero in Eq. (8) can be excluded from computations so that the efficiency of prediction can be further improved using the scarcity of prediction block F_i . Considering that the values of coefficients in high frequency band are almost zero, it is necessary to redistribute the coefficients of F_i in the sequence of zigzag scanned order as

$$F_{pred} = \sum_{m=0}^3 \sum_{n=0}^3 \sum_{i=0}^3 F_i(m, n) V_i(m) H_i(n) =$$

$$F_0(0,0) V_0(0) H_0(0) + F_1(0,0) V_1(0) H_1(0) +$$

$$F_2(0,0) V_2(0) H_2(0) + F_3(0,0) V_3(0) H_3(0) +$$

$$F_0(0,1) V_0(0) H_0(1) + F_1(0,1) V_1(0) H_1(1) +$$

$$F_2(0,1) V_2(0) H_2(1) + F_3(0,1) V_3(0) H_3(1) +$$

$$F_0(1,0) V_0(1) H_0(0) + F_1(1,0) V_1(1) H_1(0) +$$

$$F_2(1,0) V_2(1) H_2(0) + F_3(1,0) V_3(1) H_3(0) + \dots +$$

$$F_0(3,3) V_0(3) H_0(3) + F_1(3,3) V_1(3) H_1(3) +$$

$$F_2(3,3) V_2(3) H_2(3) + F_3(3,3) V_3(3) H_3(3). \quad (9)$$

The recursive equation derived from Eq. (9) provides an efficient way to calculate F_i by adjusting non-zero factor of F_i . Furthermore, the computational complexity can be efficiently reduced according to the block size and the number of non-zero factors of F_i . From Eq. (9), the operation of outer product of V_i and H_i needs to be performed only with non-zero factors of F_i , and Eq. (9) can be expressed in a recursive form as

$$F_{pred} = F_{pred}^{k-1} + F_i(m, n) V_i(m) H_i(n). \quad (10)$$

Here, F_{pred}^0 is a 4×4 zero matrix and k increases from one to the total number of non-zero DCT factors in the block.

3. Shifting Matrix for Sub-Pixel Accuracy ME

The spatial-domain ME supporting sub-pixel accuracy utilizes interpolation algorithms which often up-sample more than four times of its samples. Therefore, the performance of motion estimation depends on the interpolation algorithm applied to it. For example, the performance of bilinear interpolation incorporated in the ME with half-pixel accuracy is degraded, compared to the higher-order interpolation algorithms [9]. Meanwhile, the higher-order interpolation algorithms, such as b-spline interpolation or cubic interpolation, each of which is given in Eq. (11) and Eq. (12), respectively, provide better prediction performance than the bilinear interpolation at the cost of large computational complexity. Accordingly, H.264 can enhance the image quality by utilizing a ME with quarter-pixel accuracy; however it exacerbates the computational complexity at the same time.

$$f(x) = \begin{cases} \frac{1}{2}|x|^3 - |x|^2 + \frac{2}{3}, & 0 \leq |x| < 1 \\ -\frac{1}{6}|x|^3 + |x|^2 - 2|x| + \frac{4}{3}, & 1 \leq |x| < 2 \\ 0, & 2 \leq |x| \end{cases} \quad (11)$$

$$f(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1, & 0 \leq |x| < 1 \\ a|x|^3 + 5a|x|^2 - 8a|x| + 4a, & 1 \leq |x| < 2 \\ 0, & 2 \leq |x| \end{cases} \quad (12)$$

This section presents a shifting matrix algorithm supporting quarter-pixel accuracy of ME and proposes a transform-domain ME with b-spline or cubic interpolation schemes, with which the performance can be enhanced compared to the bilinear interpolation. In order to apply b-spline or cubic interpolation, the shifting matrix should be represented as follows.

$$h_1 = h_3 = f(d)D_{[d]} + f(1-d)D_{[d]} + f(1+d)D_{[d]-1} + f(2-d)D_{[d]+1} \quad (13)$$

$$h_0 = h_2 = f(d)D_{4-[d]}^t + f(1-d)D_{4-[d]}^t + f(1+d)D_{4-[d]+1}^t + f(2-d)D_{4-[d]-1}^t \quad (14)$$

$$v_0 = v_1 = h_0^t, \quad v_2 = v_3 = h_1^t \quad (15)$$

Considering the quarter-pixel accuracy of ME, $d = \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}\}$. If $n < 0$, D_n and D_n^t use 4×4 zero matrices, otherwise if $n > 4$, $D_n = D_{8 \text{ mod } n}^t$ and $D_n^t = D_{8 \text{ mod } n}$. In Eqs. (13) and (14), $[d]$ and $[d]^t$ denote the rounding-up and rounding-down values of d , respectively.

In the case of bilinear interpolation, the frequency information in both sides of interpolated location is used to estimate the motion at the sub-pixel accuracy. On the other hand, in the case of b-spline or cubic interpolation, the frequency information in four locations is required. Therefore, when $|d| < 1$ or $|d| > 3$, the frequency information beyond the boundary of reference block can be utilized. However, in order to obtain frequency information in a specific location from the block converted to the transform domain, all of the block information including the specific location is required, which causes an unnecessary access of memory. To resolve this problem, the vertical or horizontal intra prediction mode is adopted in predicting a specific location [10].

For instance, if the current frame f_{curr} moves in y-axis at the amount of $-4 < d < -3$, the information beyond boundary of the search area is required and the problem of memory allocation happens. However, if the intra prediction mode is applied, the information beyond boundary can be estimated by duplicating information in the search area upwards as indicated in Fig. 5. In this way, the problem of unnecessary memory allocation can be effectively avoided. The following Eqs. (16) and (17) show the vertical shifting matrix in the ME with quarter-pixel accuracy for the example illustrated in Fig. 5.

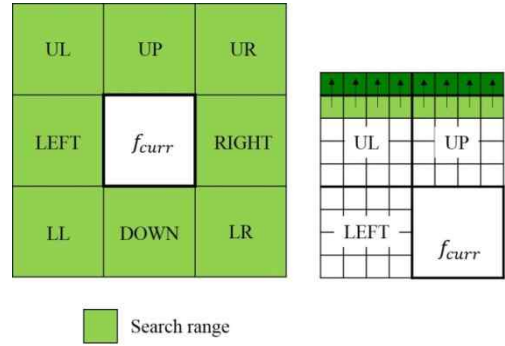


Fig. 5. Estimation of information beyond boundary using intra prediction mode

Table 1. Comparison of computational complexity in spatial-domain ME and transform-domain ME

	Spatial-domain ME	Transform-domain ME
Addition	$16p^2(32N^2 - 1) + (45Np)^2 + 2N^2(N - 1)$	$16p^2(2N^2 - 1) + 16N^3p^2$
Multiplication	$(60Np)^2 + 2N^3$	$(16Np)^2N$

$$v_0 = v_1 = \begin{bmatrix} f(1+d) + f(d) & f(1-d) & f(2-d) & 0 \\ f(1+d) & f(d) & f(1-d) & f(2-d) \\ 0 & f(1+d) & f(1-d) & f(2-d) \\ 0 & 0 & f(1+d) & f(1-d) \end{bmatrix} \quad (16)$$

$$v_2 = v_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ f(2-d) & 0 & 0 & 0 \\ f(1-d) & f(2-d) & 0 & 0 \end{bmatrix} \quad (17)$$

It is worthwhile to note that the proposed mechanism does not require IDCT and alleviates the computational complexity by utilizing the recursive equation. Table 1 compares the number of addition and multiplication operations used in the spatial-domain ME and the transform-domain ME when the block size is $N \times N$ and the search area is p . For the typical values of $N = 4$ and $p = 8$, the transform-domain ME decreases the number of addition and multiplication operations by about 27 and 3.5 times, respectively, compared to the spatial-domain ME, which confirms the outstanding decrease of computational complexity due to the transform-domain approach.

In order to compare the computational complexities of spatial-domain ME and transform-domain ME, two indices of γ^+ and γ^* are introduced as

$$\gamma^+ = \frac{N_{SD}^+}{N_{SD}^*}, \quad \gamma^* = \frac{N_{TD}^*}{N_{SD}^*},$$

where, N_{SD}^+ and N_{TD}^+ are the numbers of addition operations in the spatial-domain ME and transform-domain ME, respectively, and N_{SD}^* and N_{TD}^* are the corresponding numbers of multiplication operations. Fig. 6(a) and 6(b) show γ^+ and γ^* when N ranges from 2 to 8 and p ranges from 2 to 10, respectively. In these ranges of N and p , γ^+

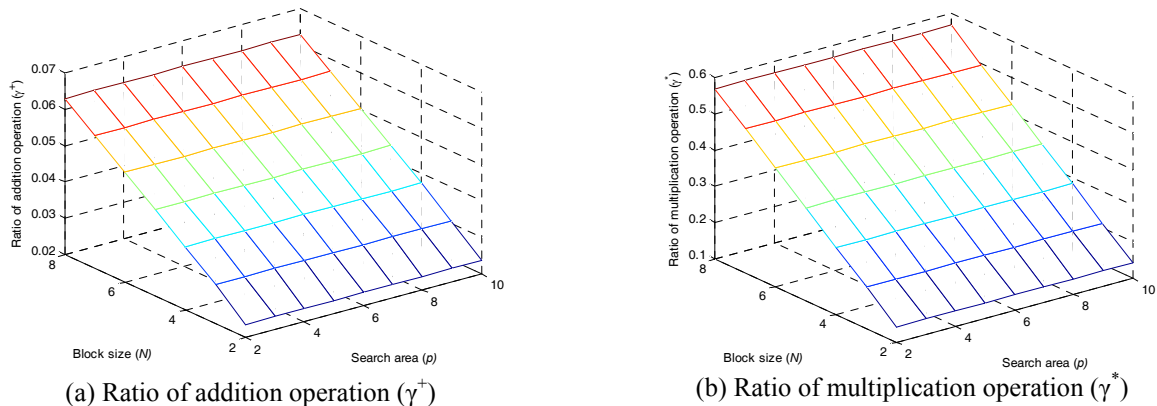


Fig. 6. Relative computational complexity of transform-domain ME to spatial-domain ME

Table 2. Performance comparison of several motion estimation algorithms for akiyo sequence

QP	PSNR (dB)							BPF (Kbit/frame)						
	H.264	CSD	BSD	CTD	BTD	HCL	HBL	H.264	CSD	BSD	CTD	BTD	HCL	HBL
3	62.18	62.10	62.15	61.95	61.91	61.92	61.91	129.2	127.9	128.5	119.9	117.0	118.5	117.1
6	56.80	56.86	56.81	56.46	56.57	56.47	56.57	147.5	144.3	145.8	105.9	102.1	103.9	102.2
9	53.27	53.36	53.31	53.63	53.78	53.67	53.79	120.4	115.5	118.3	91.3	85.5	89.0	85.5
12	50.86	51.01	50.92	51.42	51.77	51.56	51.78	96.2	90.2	93.8	76.0	65.3	72.5	65.3
15	49.22	49.38	49.28	49.60	49.93	49.76	49.93	56.9	52.5	54.6	54.5	46.8	51.5	46.9
18	47.44	47.64	47.51	47.73	48.20	47.96	48.20	37.4	34.4	36.2	37.9	32.0	34.8	32.1
21	45.88	46.07	45.96	46.06	46.55	46.27	46.55	22.7	21.4	22.1	25.0	22.1	23.2	22.1
24	44.05	44.28	44.14	44.25	44.85	44.53	44.83	16.2	15.2	15.6	18.0	16.0	16.7	16.0
27	42.41	42.77	42.55	42.49	43.24	42.90	43.23	11.6	10.9	11.3	13.3	12.0	12.2	12.0
30	40.52	40.98	40.67	40.52	41.28	41.05	41.28	9.4	8.9	9.2	10.9	10.0	10.2	10.1

lies between 0.024 and 0.063, while γ^* lies between 0.14 and 0.57; that is, the transform-domain ME remarkably decreases the numbers of addition and multiplication operations by about 16~42 times and 2~7 times, respectively, compared to the spatial-domain ME. Moreover, as shown in Fig. 6, both γ^+ and γ^* almost linearly increase with respect to the increase of block size N , but they are hardly affected by search area p .

4. Simulation and Discussion

In this section, the performances of several ME algorithms are evaluated and compared via simulations. In the simulations, H.264/AVC video coder is used, the quarter-pixel accuracy ME is considered, and the sum of absolute difference is used as a criterion to determine the motion vector. The performance is evaluated with two indices, PSNR and the number of bits per frame (BPF), which are widely used to measure video quality and compression efficiency. The performance is compared by considering the following several ME algorithms.

- H.264: This is a baseline algorithm as standardized by ITU-T and ISO. It is the most typical spatial-domain ME algorithm and uses a 4-tap FIR filter as the interpolation scheme.

- CSD/BSO: These two are the spatial-domain ME algorithms with the cubic and b-spline interpolation schemes, respectively.
- CTD/BTD: These two are the proposed transform-domain ME algorithms with the cubic and b-spline interpolation schemes, respectively.
- HCL/HBL: These two correspond to the enhanced versions of CTD and BTD, respectively. In order to further mitigate the computational complexity, a hybrid approach is adopted, i.e., the bilinear interpolation is applied to half-pixel and the cubic (HCL) or b-spline (HBL) interpolation is applied to quarter-pixel.

Note that H.264, CSD, and BSD are the spatial-domain ME algorithms while the others are the transform-domain ME algorithm as proposed in this paper.

Table 2 and 3 demonstrate the performance of several ME algorithms using akiyo and football sequences in CIF format with different value of quantization parameter (QP), respectively. In these tables, the values in bold type indicate the highest PSNR and lowest BPF for each QP. In akiyo sequence (see Table 2), the transform-domain ME algorithms.

However, when $QP \geq 21$, the transform-domain algorithms require slightly higher bit-rate than the spatial-domain algorithms, which results from the increase of quantization error with respect to the increase of QP.

Table 3. Performance comparison of several motion estimation algorithms for football sequence

QP	PSNR (dB)							BPF (Kbit/frame)						
	H.264	CSD	BSD	CTD	BTD	HCL	HBL	H.264	CSD	BSD	CTD	BTD	HCL	HBL
3	61.52	61.56	61.53	61.53	61.62	61.54	61.64	387.3	383.8	385.9	387.4	385.5	382.3	385.2
6	55.90	55.90	55.89	55.93	55.94	55.95	55.92	343.1	339.2	341.3	339.1	336.4	333.9	335.9
9	52.98	52.98	52.96	53.00	52.99	52.98	52.99	301.1	296.8	299.1	295.3	290.9	289.4	290.4
12	50.39	50.40	50.39	50.49	50.51	50.49	50.51	257.9	253.0	256.0	249.3	243.7	243.6	243.1
15	47.85	47.85	47.84	48.00	48.07	48.04	48.06	208.7	202.9	205.8	203.9	198.5	198.5	198.3
18	45.06	45.13	45.08	45.33	45.45	45.37	45.45	170.0	163.7	167.7	163.6	157.0	157.0	156.7
21	42.55	42.68	42.58	42.86	43.08	42.95	43.08	131.4	124.7	128.9	127.9	121.1	121.1	120.8
24	39.90	40.07	39.93	40.33	40.65	40.48	40.64	99.1	93.1	96.9	97.3	89.6	89.6	89.3
27	37.66	37.89	37.71	38.07	38.45	38.27	38.45	72.8	67.1	70.9	73.5	65.9	65.9	65.7
30	35.43	35.73	35.53	35.79	36.24	36.10	36.25	50.4	46.2	49.1	53.4	45.7	45.7	45.4

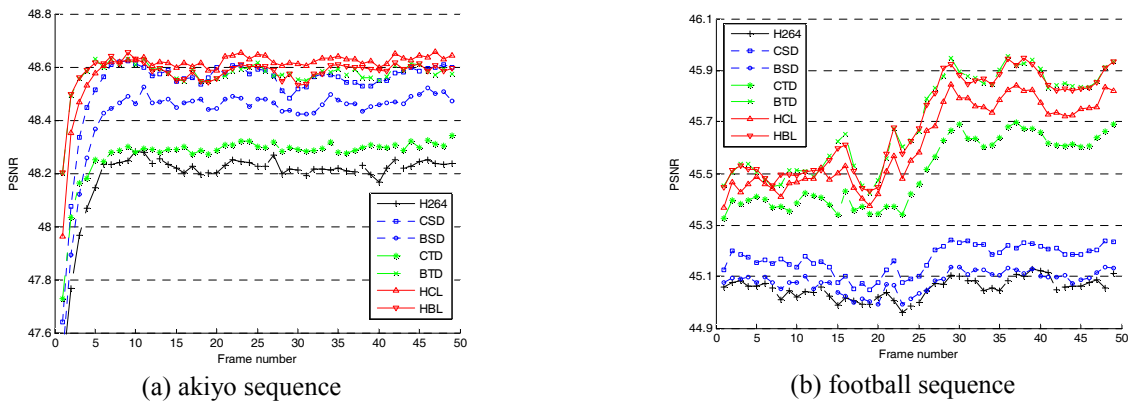


Fig. 7. Performance comparison of motion estimation in spatial domain and transform domain (QP=18)

In the case of football sequence, as shown in Table 3, the transform-domain algorithms have higher PSNR than the spatial-domain algorithms, regardless of QP values. In terms of BPF, CTD requires more bits than the spatial-domain algorithms for some specific values of QP (e.g., QP=3 and QP≥21). This is because the quantization errors are accumulated when estimating locations at half-pixel and quarter-pixel. When estimating the location of certain sub-pixels such as 1.25, 1.5, 1.75, 3.25, 3.5 and 3.75 by using the shifting matrix, some reference locations are beyond the boundary and are forced to be duplicated as shown in Fig. 5. Interpolating these locations causes the estimation errors, leading to the decrease of PSNR. This problem can be effectively dealt with the hybrid algorithms, HCL and HBL. By applying the differentiated interpolation scheme in half-pixel location (bilinear interpolation) and quarter-pixel location (cubic interpolation in HCL and b-spline interpolation in HBL), the quantization error is reduced, which contributes to the increase of PSNR accordingly.

Fig. 7(a) and 7(b) show the values of PSNR for 50 frames with akiyo and football sequences, respectively. Here, the value of QP is fixed to 18. As shown in Fig. 7, the transform-domain algorithms achieve higher PSNR than the spatial-domain algorithms, on the whole. Also, it is shown that HCL outperforms the others for the most frames in the case of akiyo sequence (see Fig. 7(a)), while HBL and BTD attain the highest value of PSNR in the case of football sequence (see Fig. 7(b)). The value of PSNR is

high in the order of CSD, BSD, and H.264 among the spatial-domain algorithms, while it is high in the order of the hybrid algorithms (HCL and HBL), BTD and CTD among the transform-domain algorithms. When the interpolation is performed in the spatial domain, the surrounding pixels of target location are used to produce an interpolated pixel; however, the proposed transform-domain algorithms use the DCT coefficient instead. In the case of block-based 2D DCT, most of the information in the block is used to obtain the specific frequency component. This is the reason why the performance of specific spatial-domain ME does not retain a consistently in the transform domain.

5. Conclusion

This paper proposed the low-complexity transform-domain ME algorithm supporting quarter-pixel accuracy. By utilizing the shifting matrix in the transform domain, the proposed algorithm can attain high quality of video and high efficiency of compression, while reducing the computational complexity substantially. The simulation results verified that the proposed algorithm not only provides higher PSNR but also requires smaller number of bits per frame compared to the conventional ME algorithms in the spatial domain. Moreover, due to the utilization of recursive equation and lack of IDCT block, the proposed

transform-domain ME algorithm significantly relieved the computational complexity, which is the critical problem of the spatial-domain ME algorithms. As a result, the proposed algorithm is quite suitable for the real-time or streaming video service in mobile devices or embedded systems that have the limited capability of computing power, memory space, battery lifetime, and communication bandwidth. The proposed algorithm is also applicable to the next-generation video compression standard of H.265 by revising the shifting matrix to cope with the variable block size from 4×4 to 32×32 . The gain of compression efficiency that can be achieved by the proposed algorithm is further intensified with a large block size such as 32×32 , by utilizing the low-complexity recursive equation [11].

Acknowledgments

This work was supported by the research program of Dongguk University.

References

- [1] S. Agha and V. M. Dwyer, "Algorithm and VLSI architectures for MPEG-4 motion estimation," *Electronic System and Control Division Research*, 2003.
- [2] J. Oh, "Improved sub-block matching algorithm," *Journal of Korean Institute of Communications and Information Sciences*, vol. 33, no. 7, pp. 628-633, Jul. 2010.
- [3] Y. Keller and A. Averbuch, "Fast gradient methods based on global motion estimation for video compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 4, pp. 300-309, Apr. 2003.
- [4] Q. H. Nguyen, V. A. Nguyen, C. V. Trinh, K. Q. Dinh, Y. Park, and B. Jeon, "Block-based transform-domain measurement coding for compressive sensing of images," *Journal of Korean Institute of Communications and Information Sciences*, vol. 39A, no. 12, pp. 746-755, Dec. 2014.
- [5] C. M. Yang and K. C. hung, "Embedded image compression scheme using rate-distortion optimized block coding of wavelet coefficients," *Journal of Korean Institute of Communications and Information Sciences*, vol. 39A, no. 11, pp. 625-636, Nov.2014.
- [6] J. Song and B. Yeo, "A fast algorithm for DCT-domain inverse motion compensation based on shared information a macroblock," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 5, pp. 767-775, Aug. 2000.
- [7] R. Kleihorst and F. Caberera, "Implement of DCT-domain motion estimation and compensation," in *Proceedings of IEEE Workshop on Signal Processing Systems (SIPS)*, pp. 53-62, Oct. 1998.
- [8] R.H.J.M. Plompen, B. F. Schuurink, and J. Biemond,

"A new motion-compensated transform coding scheme," *IEEE Transactions on Acoustics, Speech and Signal Process*, vol. 1, pp. 371-374, 1985.

- [9] M. Kang, J. Heo, and C. Ryu, "Half-pixel accuracy motion estimation algorithm in the transform domain for H.264," *Journal of Korean Institute of Communications and Information Sciences*, vol. 33, no. 11, pp. 917-924, Nov. 2011.
- [10] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560-576, Jul. 2003.
- [11] D. Yoon and Y. Ho, "Fast mode decision method for HEVC in depth video," *Journal of Korean Institute of Communications and Information Sciences*, vol. 37, no. 1A, pp. 51-56, Jan. 2012.



Chul Ryu He received his M.S and Ph.D degrees in electrical engineering in 1991 and 1997, respectively, both from the Polytechnic Institute of New York University. From 1998 to 1999, he served as a research engineer of advanced wireless technologies lab at LG Electronics. He has been a pro-

fessor of the Department of Information and Communication Engineering, Dongguk University, Seoul since 1999. His research area includes visual communication, super-resolution reconstruction, and real-time hardware design for video codec.



Jae-Young Shin She received Master's degree in Information and Communication Engineering from Dongguk University. Her research interests are digital image processing and compression.



Eun-Chan Park He received B.S, M.S, and Ph.D degrees from the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea, in 1999, 2001, 2006, respectively. From 2006 to 2008, he was a senior engineer with Samsung Electronics, Korea. He is currently an

associate professor in the Department of Information and Communication Engineering, Dongguk University, Seoul since 2006. His research interests include network performance analysis, resource allocation and MAC protocol design of wireless networks.