Journal of Marine Science and Technology

Volume 18 | Issue 1 Article 11

FAST BLOCK MATCHING USING MINKOWSKI'S INEQUALITY

Jim Zong-Chang Lai
Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan, R.O.C

Yi-Ching Liaw
Department of Computer Science and Information Engineering, Nanhua University, Chiayi, Taiwan, R.O.C., ycliaw@mail.nhu.edu.tw

Shih-An Fong
Information and Communications Research Labs, Industrial Technology Research Institute, Hsinchu, Taiwan, R.O.C.

Follow this and additional works at: https://jmstt.ntou.edu.tw/journal

 Part of the Electrical and Computer Engineering Commons

Recommended Citation

Lai, Jim Zong-Chang; Liaw, Yi-Ching; and Fong, Shih-An (2010) "FAST BLOCK MATCHING USING MINKOWSKI'S INEQUALITY," *Journal of Marine Science and Technology*: Vol. 18 : Iss. 1 , Article 11.
DOI: 10.51400/2709-6998.1869
Available at: https://jmstt.ntou.edu.tw/journal/vol18/iss1/11

This Research Article is brought to you for free and open access by Journal of Marine Science and Technology. It has been accepted for inclusion in Journal of Marine Science and Technology by an authorized editor of Journal of Marine Science and Technology.

FAST BLOCK MATCHING USING MINKOWSKI'S INEQUALITY

Jim Zong-Chang Lai*, Yi-Ching Liaw**, and Shih-An Fong***

Key words: motion estimation, block matching, sum pyramid.

ABSTRACT

In this paper, we present a fast block matching algorithm by making use of the Minkowski's inequality. Combined with other fast block matching algorithms, our method can further reduce the computational complexity significantly with little PSNR degradation. Compared to the full search block matching algorithm, our approach can reduce the computing time by a factor of 6.4 to 21.5. Compared to MSEQ, which is a well-known block matching algorithm preserving global optimality, our method can further reduce the corresponding computing time in average by 32.4%. The combination of our algorithm and the predictive search area approach for block motion estimation can achieve the more reduction of computing time with little PSNR degradation.

I. INTRODUCTION

Motion estimation using block matching has been widely used for video coding, such as H.264 and MPEG standards [2, 22] and for video applications [8, 18]. In a block matching algorithm (BMA), the current frame of an image sequence is divided into nonoverlapping blocks of $N \times N$ pixels. For each template block in the current frame, the best matched block within a search window of size $(2W + 1) \times (2W + 1)$ in the previous frame is determined, where W is the maximum allowed displacement. The position difference between a template block in the current frame and the best matched block in the previous frame is called the motion vector.

The criterion most commonly used by block matching algorithms is the sum of absolute difference (SAD), which is defined below, between a template block at position (x, y) in the current frame I_t and the candidate block at position $(x + \hat{u}, y + \hat{v})$ in the previous frame I_{t-1}.

$$\mathrm{SAD}(\hat{u}, \hat{v}) = \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \left| g_t(x+i, y+j) - g_{t-1}(x+\hat{u}+i, y+\hat{v}+j) \right| \quad (1)$$

where $g_t(., .)$ is the gray value of a pixel in the current frame and $g_{t-1}(., .)$ is the gray level of a pixel in the previous frame. In this paper, the motion vector (u, v) is defined as follows:

$$(u, v) = \arg \min_{(\hat{u}, \hat{v}) \in S} \mathrm{SAD}(\hat{u}, \hat{v}) \quad (2)$$

where $S = \{(\hat{u}, \hat{v}) \mid -W \le \hat{u}, \hat{v} \le W$ and $(x + \hat{u}, y + \hat{v})$ is a valid pixel position in $I_{t-1}\}$.

The full search block matching algorithm (FSBMA), which determines the best matched block through calculating the SAD for all positions in the search window, is the simplest among the available block matching algorithms. Although FSBMA can obtain the global optimal result, however it has very intensive computations. To overcome this problem, many fast algorithms are developed [3, 5, 7, 8, 9, 11, 12, 13, 14, 16, 23]. Some well known algorithms are the three-step search (TSS) [14], new three-step search (NTSS) [16], one-dimensional search [3, 11], cross search [7], one-at-a-time search [12, 23], hierarchical search [9, 10], block-based gradient descent search [22], fast search with predictive searching area [5], and multi-resolution block matching algorithm (MRBMA) [13]. Another approach for motion estimation is called the partial-matching error technique, which can obtain the same SAD as the FSBMA [1, 4, 6, 15, 17, 19, 21]. Compared to FSBMA, this kind of approach may reduce the computational complexity significantly. This kind of technique first calculates the complete matching error at the predicted position and uses it as the initial minimum matching error (the initial minimum SAD). For each other search position, some forms of partial matching errors are calculated. The calculation of the complete SAD can be avoided, if the partial error is larger than the minimum SAD. Otherwise, the complete matching error (the complete SAD) is calculated and the minimum SAD is updated if the newly calculated matching error is smaller than the original one.

In this paper, a fast block matching algorithm will be developed. This algorithm uses Minkowsk's inequality [3, 15, 21] to eliminate those impossible candidate blocks, which

Paper submitted 06/03/08; revised 02/20/09; accepted 02/25/09. Author for correspondence: Yi-Ching Liaw (e-mail: ycliaw@mail.nhu.edu.tw).

**Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.*

***Department of Computer Science and Information Engineering, Nanhua University, Chiayi, Taiwan, R.O.C.*

****Information and Communications Research Labs, Industrial Technology Research Institute, Hsinchu, Taiwan, R.O.C.*

cannot be rejected using available rejecting criteria, and obtain the same SAD as the full search block matching algorithm. Combined with other fast block matching algorithm, such as fast search with predictive searching area [5], our method will reduce the computing time significantly with little PSNR degradation. Our proposed method is different from the method presented in references [15] and [21], which also used the pyramid structure and the Minkowski's inequality to reject impossible blocks. The major difference is that we use a different rejection criterion to reject impossible blocks and a different way of choosing the initial block. Moreover, the proposed method is also extended to speed up the block searching speed by combining with other fast block matching algorithm.

This paper is organized as follows. We present our concepts and algorithm in section II. In section III, the combination of our algorithm and a predictive search area approach for block motion estimation (PSAFBME) [5] is presented. Experimental results are given in section IV and conclusions are presented in section V.

II. BLOCK MATCHING ALGORITHM USING ELIMINATION CRITERIA

In this section, we will present our block matching algorithm and the corresponding elimination criteria. Our method first selects an initial candidate block to calculate its SAD and uses this SAD as the minimum one. A modified version of pyramidal lower bound [15] is developed and is used to reject unlikely candidate blocks.

1. Principles of the Algorithm

Assume that the current frame I_t is divided into nonoverlapping blocks of $N \times N$ pixels with $N = 2^n$. For a block X, the pyramid of X can be defined as a set of blocks $\{X^0, X^1, \ldots, X^n\}$ with X^m being a reduced resolution version of X, where $m = 1, 2, \ldots, n$. It is noted that X^m has $2^m \times 2^m$ pixels. The value of a pixel $X^{m-1}(k, l)$ on the $(m\text{-}1)$ level can be obtained from *4* pixels $X^m(2k, 2l)$, $X^m(2k, 2l + 1)$, $X^m(2k + 1, 2l)$, and $X^m(2k + 1, 2l + 1)$ on the m level. That is,

$$X^{m-1}(k, l) = X^m(2k, 2l) + X^m(2k, 2l + 1) + X^m(2k + 1, 2l) + X^m(2k + 1, 2l + 1) \quad (3)$$

The SAD of X^m and Y^m is denoted as $SAD^m{}_{X,Y}$, which can be calculated by the following equation:

$$SAD^m{}_{X,Y} = \sum_{k=0}^{M-1}\sum_{l=0}^{M-1}\left|X^m(k,l) - Y^m(k,l)\right| \quad (4)$$

where $M = 2^m$, and $X^m(k, l)$ and $Y^m(k, l)$ represent the values of the (k, l)th pixels of X^m and Y^m, respectively. Let

$$\begin{aligned} PSAD^m{}_{X,Y}(k,l) = {} & |X^{m+1}(2k, 2l) - Y^{m+1}(2k, 2l)| \\ & + |X^{m+1}(2k + 1, 2l) - Y^{m+1}(2k + 1, 2l)| \\ & + |X^{m+1}(2k, 2l + 1) - Y^{m+1}(2k, 2l + 1)| \\ & + |X^{m+1}(2k + 1, 2l + 1) \\ & - Y^{m+1}(2k + 1, 2l + 1)| \end{aligned} \quad (5)$$

It is noted that $PSAD^m(k, l)$ represents the sum of absolute difference for *8* pixels on the $(m + 1)$th level, which represents the higher resolution of $X^m(k, l)$ and $Y^m(k, l)$. Let

$$\begin{aligned} TPSAD^m{}_{X,Y}(kk) = {} & \sum_{k=0}^{k1-1}\sum_{l=0}^{M-1} PSAD^m{}_{X,Y}(k,l) \\ & + \sum_{l=0}^{l1} PSAD^m{}_{X,Y}(k1,l) \end{aligned} \quad (6)$$

where $M = 2^m$, $k1$ = the quotient of $kk/2^m$, $l1$ = the remainder of $kk/2^m$, and $0 \le kk < 2^m \times 2^m$. From Minkowski's inequality, we can easily conclude that

$$PSAD^m{}_{X,Y}(k, l) \ge |X^m(k, l) - Y^m(k, l)| \quad (7)$$

Let $MSAD^m{}_{X,Y}(kk)$ be defined by

$$\begin{aligned} MSAD^m{}_{X,Y}(kk) = {} & TPSAD^m{}_{X,Y}(kk) \\ & + \sum_{k=k1+1}^{M-1}\sum_{l=0}^{M-1} |X^m(k, l)\text{-}Y^m(k, l)| \\ & + \sum_{l=l1+1}^{M-1} |X^m(k1, l)\text{-}Y^m(k1, l)| \end{aligned} \quad (8)$$

where $M = 2^m$, $k1$ = the quotient of $kk/2^m$, $l1$ = the remainder of $kk/2^m$. $MSAD^m{}_{X,Y}(kk)$ represents the sum of absolute gray level differences for some pixels from X^m and Y^m and others from X^{m+1} and Y^{m+1}.

From (5) to (8), we can conclude that

$$\begin{aligned} MSAD^m{}_{X,Y}(kk + 1) = {} & MSAD^m{}_{X,Y}(kk) - |X^m(k1, l1)\text{-}Y^m(k1, l1)| \\ & + PSAD^m{}_{X,Y}(k1, l1) \end{aligned} \quad (9)$$

Where $k1$ = the quotient of $(kk+1)/2^m$, and $l1$ = the remainder of $(kk + 1)/2^m$. It is noted that $MSAD^m{}_{X,Y}(2^m \times 2^m - 1) = SAD^{m+1}{}_{X,Y}$. Let X and Y be two blocks of $2^n \times 2^n$ pixels. Denote the SAD of X and Y as $SAD_{X,Y}$. It is noted that $SAD_{X,Y} = SAD^n{}_{X,Y} = MSAD^{n-1}{}_{X,Y}(2^{n-1} \times 2^{n-1} - 1)$. Using Minkowski's inequality to (4) and (8), we can find that

$$\begin{aligned} SAD_{X,Y} = {} & MSAD^{n-1}{}_{X,Y}(2^{n-1} \times 2^{n-1} - 1) \\ & \ge MSAD^{n-1}{}_{X,Y}(2^{n-1} \times 2^{n-1} - 2) \ge \ldots \ge MSAD^{n-1}{}_{X,Y}(0) \\ & \ge SAD^{n-1}{}_{X,Y} \\ & = MSAD^{n-2}{}_{X,Y}(2^{n-2} \times 2^{n-2} - 1) \ge \ldots \ge MSAD^1{}_{X,Y}(3) \\ & \ge MSAD^1{}_{X,Y}(2) \ge \ldots \ge MSAD^1{}_{X,Y}(0) \ge SAD^1{}_{X,Y} \\ & = MSAD^0{}_{X,Y}(0) \ge SAD^0{}_{X,Y} \end{aligned} \quad (10a)$$

The above expression can also be represented as

$$\mathrm{MSAD}^{l}_{X,Y}(i) \geq \mathrm{MSAD}^{l-1}_{X,Y}(j) \text{ and } \mathrm{MSAD}^{l}_{X,Y}(i) \geq \mathrm{MSAD}^{l}_{X,Y}(i-1),$$
$$0 \leq l < n,\ 0 \leq i < 2^{l} \times 2^{l}, \text{ and } 0 \leq j < 2^{l-1} \times 2^{l-1} \quad (10b)$$

With the above results in hand, we can describe our algorithm as follows. Our algorithm first constructs the sum pyramid of the previous frame using the method present in reference [16]. For an image frame of $W \times H$ pixels, the computational complexity of constructing the sum pyramid is $4(2W-1)(H-1)$ for $N = 16$, where N is the block size in I_{t-1}. For a template block X in I_t, an initial motion vector $(\hat{u}, \hat{v})$ is estimated using the spatial correlation between two frames. The estimation of $(\hat{u}, \hat{v})$ will be discussed in the next subsection. The SAD between block X and the block with displacement $(\hat{u}, \hat{v})$ in frame I_{t-1} is calculated and this value is stored as the minimum SAD, $\mathrm{SAD}_{\min}$ For other candidate block Y in frame I_{t-1}, $\mathrm{SAD}^{0}_{X,Y}$ is determined first. If $\mathrm{SAD}^{0}_{X,Y}$ is larger than $\mathrm{SAD}_{\min}$, the candidate block Y is rejected; otherwise $\mathrm{SAD}^{1}_{X,Y}$ is computed. If $\mathrm{SAD}^{1}_{X,Y}$ is greater than $\mathrm{SAD}_{\min}$, block Y is eliminated. Otherwise $\mathrm{MSAD}^{1}_{X,Y}(0)$ is calculated and check whether it is larger than $\mathrm{SAD}_{\min}$. If $\mathrm{MSAD}^{1}_{X,Y}(0)$ is larger than $\mathrm{SAD}_{\min}$, then block Y is rejected; otherwise If $\mathrm{MSAD}^{1}_{X,Y}(1)$ is calculated. This process is repeated until block Y is rejected, or the bottom level of the sum pyramid of Y is reached and $\mathrm{SAD}_{X,Y}$ is calculated. If $\mathrm{SAD}_{X,Y}$ is greater than $\mathrm{SAD}_{\min}$, then block Y is rejected; otherwise $\mathrm{SAD}_{\min}$ is replaced by $\mathrm{SAD}_{X,Y}$. This technique can eliminate many candidate blocks without calculating their SADs.

2. Initial Motion Vector Estimation

The initial motion vector (MV) can be estimated through using spatial and temporal correlations of motion vectors. The idea is that select a set of candidate motion vectors from spatially and/or temporally neighbors and choose the best one as the initial motion vector.

Let $\mathrm{TB}_{i,j}$ be a template block of $N \times N$ pixels at the (i, j)th position of the current frame I_t and $\mathrm{SB}_{k,l}$ be a candidate block at the (k, l)th position of the previous frame I_t. Denote the motion vector of block $\mathrm{TB}_{i,j}$ as $(u, v)_{i,j}$. Let the sum pyramids of $\mathrm{TB}_{i,j}$ and $\mathrm{SB}_{k,l}$ as $\{\mathrm{TB}^{0}_{i,j}, \mathrm{TB}^{1}_{i,j}, .., \mathrm{TB}^{n}_{i,j}\}$ and $\{\mathrm{SB}^{0}_{k,l}, \mathrm{SB}^{1}_{k,l}, .., \mathrm{SB}^{n}_{k,l}\}$, respectively, where $n = \log_2 N$. The SAD between $\mathrm{TB}^{m}_{i,j}$ and $\mathrm{SB}^{m}_{i,j}$ is denoted as $\mathrm{SAD}^{m}_{u,v}$, where $m = 1, 2, \ldots, n$, $u = k - i$, and $v = l - j$. Denote the four neighboring blocks of $\mathrm{TB}_{i,j}$ as TB_p $(p = 1-4)$ as shown in Fig. 1. Let the motion vector of TB_p $(p = 1-4)$ be (u_p, v_p). Similarly, denote the SAD between $\mathrm{TB}^{m}_{i,j}$ and $\mathrm{SB}^{m}_{u_p, v_p}$ as $\mathrm{SAD}^{m}_{u_p, v_p}$, $p = 1-4$. The estimated motion vector of $\mathrm{TB}_{i,j}$ is represented by $(\hat{u}, \hat{v})_{i,j}$, which is determined as

$$(\hat{u}, \hat{v})_{i,j} = \arg \min_{(u,v) \in S} \mathrm{SAD}^{1}_{u,v} \quad (11)$$

where $S = \{(u_1, v_1), (u_2, v_2), (u_3, v_3), \text{ and } (u_4, v_4)\}$.

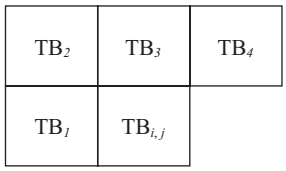

Fig. 1. A block $\mathrm{TB}_{i,j}$ and its four neighboring blocks.

At this stage, we would like to present our fast block matching algorithm using Minkowski's inequality (FBMAUMI) as follows.

(1) Input a template block TB in frame I_t and use equation (11) to determine its estimated motion vector $(\hat{u}, \hat{v})$. Set the motion vector $(u, v) = (\hat{u}, \hat{v})$.
(2) Use Eq. (1) to calculate $\mathrm{SAD}(\hat{u}, \hat{v})$ and let $\mathrm{SAD}_{\min} = \mathrm{SAD}(\hat{u}, \hat{v})$.
(3) For each block SB in the search window of size $(2W + 1) \times (2W + 1)$, use block rejection algorithm to update $\mathrm{SAD}_{\min}$ and the motion vector (u, v).
(4) Output $\mathrm{SAD}_{\min}$ and the motion vector (u, v) of TB.

The block rejection algorithm is presented below.

(1) Let the motion vector of SB with respect to TB be (u^*, v^*) and set $m = 0$.
(2) Calculate $\mathrm{SAD}^{m}_{TB,SB}$. If $\mathrm{SAD}^{m}_{TB,SB} \geq \mathrm{SAD}_{\min}$, rejected the candidate block SB and leave (u, v) and $\mathrm{SAD}_{\min}$ unchanged; otherwise set $\mathrm{MSAD}^{0}_{TB,SB}(-1) = \mathrm{SAD}^{m}_{TB,SB}$.
(3) If $m < n$, where $n = \log_2 N$, do the following procedures:
  (3a) For $i = 0$ to $(2^{m-1} \times 2^{m-1} - 1)$, calculate $\mathrm{MSAD}^{m}_{TB,SB}(i)$ using Eq. (9). If $\mathrm{MSAD}^{m}_{TB,SB}(i) \geq \mathrm{SAD}_{\min}$, rejected the candidate the block SB and leave (u, v) and $\mathrm{SAD}_{\min}$ unchanged.
  (3b) Set $m = m + 1$, $\mathrm{MSAD}^{m}_{TB,SB}(-1) = \mathrm{MSAD}^{m-1}_{TB,SB}(2^{m-1} \times 2^{m-1} - 1)$, and go to step (3).
(4) Let $(u, v) = (u^*, v^*)$ and $\mathrm{SAD}_{\min} = \mathrm{MSAD}^{n-1}_{TB,SB}(2^{n-1} \times 2^{n-1} - 1)$.

III. OMBINATION OF OUR APPROACH WITH PSAFBME

The proposed fast motion block matching algorithm can be combined with other fast algorithms to achieve further speedup. However, the guarantee of global minimum solution may be sacrificed. In this paper, we use the algorithm of predicting search area for block motion estimation [5] as an example. The predictive search area approach for block area motion estimation (PSAFBME) generates a search area SA for a template block $\mathrm{TB}_{i,j}$ in the current frame I_t. It is noted that SA in general is much smaller than the full search window of size $(2W + 1) \times (2W + 1)$. SA can be represented by

$$\mathrm{SA} = \mathrm{SA}_1 \cup \mathrm{SA}_2 \cup \mathrm{SA}_3 \cup \mathrm{SA}_4 \quad (12)$$

Where SA_k ($k = 1 - 4$) denotes the subsearach area of size $(2D + 1) \times (2D + 1)$ with center at $(i + u_k, j + v_k)$, where (u_k, v_k) is the motion vector of the block TB_k ($k = 1 - 4$) as shown in Fig. 1. It is noted that TB_k is the kth neighboring block of $TB_{i,j}$. $D = 2$ or 3 may be used to obtain little image degradation. The combination of our approach with PSAFBME is referred to as MPSAFBME in this paper. Compared to PSAFBME, MPSAFBME has the much less computational complexity. Now, we would like to present the modified predictive search area for block motion estimation (MPSAFBME) as follows.

(1) Input a template block TB in frame I_t and use Eq. (11) to determine its estimated motion vector $(\hat{u}, \hat{v})$. Set the motion vector $(u, v) = (\hat{u}, \hat{v})$.
(2) Use Eq. (1) to calculate $SAD(\hat{u}, \hat{v})$ and let $SAD_{min} = SAD(\hat{u}, \hat{v})$.
(3) For each block SB in the search area SA, use block rejection algorithm to update SAD_{min} and the motion vector (u, v).
(4) Output SAD_{min} and the motion vector (u, v) of TB.

IV. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed algorithm, six sets of video sequences (as shown in Fig. 2) are used and several motion block estimation algorithms are implemented. The first 50 frames in six video sequences are used in the simulation. Each image of the above mentioned video sequences is of size 352×240.

All computing is performed on an Intel Core 2 Duo 6600 2.4 GHz PC with 1 GB of memory. The full search block matching algorithm (FSBMA), the partial distortion elimination (PDE) [1], the successive elimination algorithm (SEA) [17], the multilevel successive elimination algorithm (MSEA) with three levels [6], the winner-update algorithm with the lower bound derived from Minkowski's inequality (WinUpMI) [4], and our proposed method (FBMAUMI) provide the global optimality. It is noted that MSEA with three levels obtains the same results as references [15] and [21] in our implementation. MPSAFBME (presented in section III), PSAFBME [5], TSS [14], and BBGDS (block-based gradient descent search) [20] sacrifice motion estimation accuracy to gain the reduction of computational complexity. The global optimality is not guaranteed by these four algorithms.

The PSNR (peak signal-to-noise ratio) and computing time are used to evaluate the performances of block matching algorithms. In this paper, PSNR is defined by

$$PSNR = 10\log_{10}(255^2/MSE) \quad (13)$$

where MSE is the mean square error between the original and estimated image frames. In this paper, all the template and candidate blocks are of size 16×16. The size of search window is 33×33. Tables 1 and 2 show the PSNRs and computing time of six video sequences for the FSBMA, PDE, SEA, MSEA, WinUpMI, and FBMAUMI methods. From Tables 1 and 2, we can find that all algorithms obtain the same PSNR for each video sequence and our proposed method FBMAUMI performs better than the other methods. Comparing to FSBMA, FBMAUMI can reduce the computing time by a factor of 6.4 to 21.5. Compared with MSEA, which is the second best, our proposed method FBMAUMI can reduce the computing time in average by 32.4%.

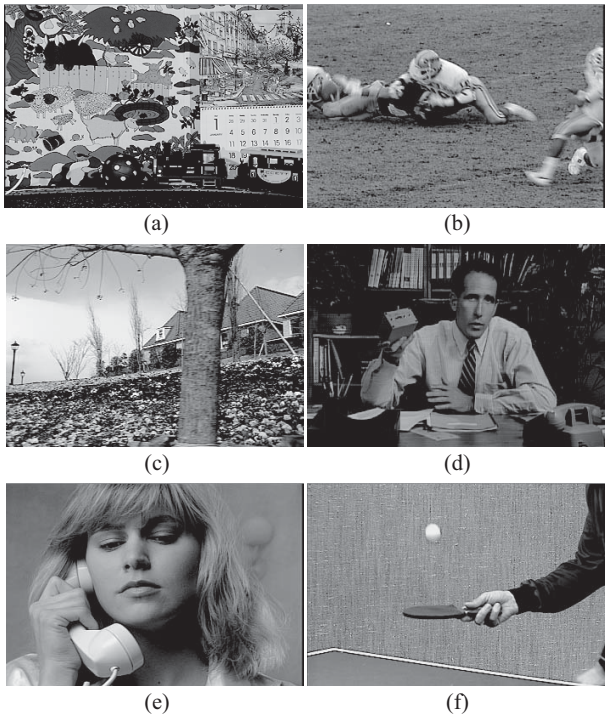

Fig. 2. The six test video sequences are (a) Mobile, (b) Football, (c) Garden, (d) Salesman, (e) Susie, and (f) Tennis.

Tables 3 and 4 give the PSNRs and computing time for MPSAFBME with $D = 2$, PSAFBME with $D = 2$, TSS, and BBGDS. From Tables 2 and 4, we can find that any one of these four algorithms obtains the lower PSNR than that of FSBMA or FBMAUMI, since these four methods sacrifice motion estimation accuracy to gain the reduction of computational complexity. From Table 3, we can find that MPSAFBME has the highest PSNR. Compared with BBGDS, MPSAFBME with $D = 2$ gives the higher PSNR with about the same computing time. From Tables 1 to 4, we may conclude that compared to FBMAUMI, MPSAFBME with $D = 2$ can reduce the computing time by a factor of about 2.86 with little PSNR degradation. Compared with PSAFBME, our proposed method can further reduce the computing time by 55.0% to 69.5%.

V. CONCLUSIONS

In this paper, we develop a fast block matching algorithm using Minkowski's inequality (FBMAUMI) to speed up motion estimation. Combining PSAFBME with our method, we

Table 1. PSNR comparison for six methods preserving global optimality.

Video Sequence	Method					
	FSBMA	PDE	SEA	MSEA	WinUpMI	FBMAUMI
Mobile	23.34	23.34	23.34	23.34	23.34	23.34
Football	22.79	22.79	22.79	22.79	22.79	22.79
Garden	23.78	23.78	23.78	23.78	23.78	23.78
Salesman	35.40	35.40	35.40	35.40	35.40	35.40
Susie	35.96	35.96	35.96	35.96	35.96	35.96
Tennis	29.07	29.07	29.07	29.07	29.07	29.07
Average	28.39	28.39	28.39	28.39	28.39	28.39

Table 2. Computing time (in seconds) comparison for six methods preserving global optimality.

Video Sequence	Method					
	FSBMA	PDE	SEA	MSEA	WinUpMI	FBMAUMI
Mobile	10.687	6.641	4.281	0.703	0.811	0.564
Football	10.827	7.297	6.455	1.735	1.438	0.969
Garden	10.660	6.562	4.421	0.922	0.952	0.581
Salesman	10.718	5.812	2.779	0.466	0.516	0.499
Susie	10.703	6.110	3.096	1.014	1.156	0.669
Tennis	10.685	7.423	7.406	2.484	2.982	1.671
Average	10.713	6.641	4.740	1.221	1.309	0.826

Table 3. PSNR comparison for four methods not preserving global optimality.

Video Sequence	Method			
	TSS	PSAFBME	BBGDS	MPSAFBME
Mobile	23.05	23.33	23.33	23.33
Football	22.01	21.66	21.55	21.66
Garden	21.80	23.66	23.42	23.66
Salesman	35.14	35.37	35.37	35.37
Susie	34.90	35.89	35.78	35.89
Tennis	27.58	27.87	27.73	27.87
Average PSNR	27.41	27.96	27.87	27.96

Table 4. Computing time (in seconds) comparison for four methods not preserving global optimality.

Video Sequence	Method			
	TSS	PSAFBME	BBGDS	MPSAFBME
Mobile	0.344	0.718	0.173	0.219
Football	0.358	0.828	0.266	0.359
Garden	0.359	0.797	0.220	0.251
Salesman	0.343	0.625	0.188	0.281
Susie	0.342	0.749	0.218	0.344
Tennis	0.358	0.780	0.266	0.280
Average time	0.351	0.750	0.222	0.289

can reduce computing time significantly with little PSNR degradation. Compared to FSBMA, our proposed method (FBMAUMI) can reduce the computing time by a factor of 6.4 to 21.5. Compared to the well-know multilevel successive elimination algorithm preserving global optimality (MSEA), our algorithm can further reduce the computing time in average by 32.4%. MPSFBME with $D = 2$ reduces the computing time of FSBMA by 96.7% to 98.0% with the average PSNR degradation of 0.43 dB.

REFERENCES

1. Bei, C. D. and Gray, R. M., "An improvement of the minimum distortion encoding algorithm for vector quantization," *IEEE Transactions on Communications*, Vol. 33, No. 10, pp. 1132-1133 (1985).
2. Bhaskaran, V. and Konstantinides, K., *Image and Image Compression Standards: Algorithms and Standards*, Dordrecht/Norwell, MA: Kluwer Academic (1997).
3. Chen, M. J., Chen, L. G., and Chiueh, T. D., "One-dimensional full search motion estimation algorithm for video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 4, No. 5, pp. 504-509 (1994).
4. Chen, Y. S., Hung, Y. P., and Fuh, C. S., "Fast block matching algorithm based on the winner-update strategy," *IEEE Transactions on Image Processing*, Vol. 10, No. 8, pp. 1212-1222 (2001).
5. Chung, K. L. and Chang, L. C., "A new predictive search area approach for fast block motion estimation," *IEEE Transactions on Image Processing*, Vol. 12, No. 6, pp. 648-652 (2003).
6. Gao, X. Q., DuanMu, C. J., and Zou, C. R., "A multilevel successive elimination algorithm for block matching motion estimation," *IEEE Transactions on Image Processing*, Vol. 9 No. 3, pp. 501-504 (2000).
7. Ghanbari, M., "The cross-serach algorithm for motion estimation," *IEEE Transactions on Communications*, Vol. 38, No. 7, pp. 950-983 (1990).
8. Hariharakrishnan, K. and Schonfeld, D., "Fast object tracking using adaptive block matching," *IEEE Transactions on Multimedia*, Vol. 7, No 5, pp. 853-859 (2005).
9. Hepper, D., "Efficiency analysis application of uncovered background prediction in a low bit rate image coder," *IEEE Transactions on Communications*, Vol. 36, No. 9, pp. 1578-1584 (1990).
10. Huang, C. L. and Hsu, C, Y., "A new compensation method for image sequence coding using hierarchical grid interpolation," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 4, No. 1, pp. 72-85 (1994).
11. Jain, J. R. and Jain, A. K., "Displacement measurement and its application in interframe coding," *IEEE Transactions on Communications*, Vol. 29, No. 12, pp. 1799-1808 (1981).
12. Kappagantula, S. and Rao, K. R., "Motion compensated interframe image prediction," *IEEE Transactions on Communications*, Vol. COMM-33, No. 9, pp. 1011-1015 (1985).
13. Kim, M. J., Lee, Y. G., and Ra, J. B., "A fast multi-resolution block matching algorithm for multiple-frame motion estimation," *IEICE Transactions on Information and Systems*, Vol. E88-D, No. 12, pp. 2819-2827 (2005).
14. Koga, T., Iinuma, K., Hirano, A., Iijima, Y., and Ishiguro, T., "Motion-compensated interframe coding for video conferencing," in *Proceeding of National Telecommunication Conference*, pp. C9.6.1-C9.6.5, New Orleans, LA (1981).
15. Lee, C. H. and Chen, L. H., "A fast motion estimation algorithm based on the block sum pyramid," *IEEE Transactions on Image Processing*, Vol. 6 No. 11, pp. 1587-1591 (1997).
16. Li, R., Zeng, B., and Liou, M. L., "A new three search algorithm for three step block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 4, No. 4, pp. 438-442 (1994).

17. Li, W. and Salari, E., "Successive elimination algorithm for motion estimation," *IEEE Transactions on Image Processing*, Vol. 14 No. 1, pp. 105-107 (1995).
18. Lin, C. F. and Tseng, C. Y., "Development of a cost effective mini autonomous underwater vehicle," *Journal of Marine Science and Technology*, Vol. 14, No. 2, pp. 119-126 (2006).
19. Lin, Y. C. and Tai, S. C., "Fast full search block-matching algorithm for motion-compensated video compression," *IEEE Transactions on Communications*, Vol. 45 No. 5, pp. 527-531 (1997).
20. Liu, L. K. and Feig, E., "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 6, No. 4, pp. 419-422 (1996).
21. Lu, J. Y., Wu, K. S., and Lin, J. C., "Fast full search in motion estimation by hierarchical use of Minkowski's inequality (HUMI)," *Pattern Recognition*, Vol. 31, No. 7, pp. 945-952 (1998).
22. Richardson, I. E. G., *H.364 and MPEG-4 Video Compression*, San Francisco, CA: John Wiley & Sons (2003).
23. Srinivasan, R. and Rao, K. R., "Predictive coding on efficient motion estimation," *IEEE Transactions on Communications*, Vol. COMM-33, No. 8, pp. 888-896 (1985).