Coarse-to-Fine Partial Distortion Search Algorithm for Motion Compensation Estimation

Article *in* International journal of innovative computing, information & control: IJICIC · September 2009

3 authors:

Chia-Hung Yeh
National Sun Yat-sen University

Ming-Te Wu
Kao Yuan University

Shiunn-Jang Chern
Tamkang University

International Journal of Innovative Computing, Information and Control
Volume **5**, Number **9**, September **2009**

pp. 2523–2530

COARSE-TO-FINE PARTIAL DISTORTION SEARCH ALGORITHM FOR MOTION ESTIMATION

Chia-Hung Yeh, Ming-Te Wu and Shiunn-Jang Chern

Center for Wireless Multimedia Communications
Department of Electrical Engineering
National Sun Yat-Sen University
Kaohsiung 80424, Taiwan
{ yeh; chern }@mail.ee.nsysu.edu.tw

Received March 2008; revised August 2008

Abstract. *This study proposes a coarse-to-fine partial distortion search algorithm for motion estimation to accurately predict the minimum distortion region. In the proposed method, the distortion behavior is first analyzed in a search window to indicate the possible occurrence region of the global minimum distortion position. Normalized partial distortion searching is then performed on the selected region, instead of the whole search window. The proposed algorithms have a low computational complexity yet do not degrade the PSNR significantly.*
Keywords: Normalized partial distortion search, Motion estimation, Block-matching distortion, Full search, Sum of absolute difference

1. **Introduction.** Many video coding standards, such as H.26X [1,2] and MPEG-X [3,4] have been proposed for various applications [5]. These coding standards adopt motion estimation to remove the temporal redundancy of the interframes. The block-matching algorithm (BMA) is widely used to find motion vectors (MV) [6]. The most straightforward BMA is the full search (FS) algorithm, which matches all possible blocks within the restricted search area, called the search window, in the previous frame to find the block with the minimum block-matching distortion (BMD) defined as the sum of absolution difference (SAD). However, the FS algorithm requires massive computations to calculate the BMD, and is thus unsuitable for real-time implementation.

Motion estimation is the key component of video coding. Fast and accurate motion estimation method is highly desired to achieve high compression ratio while maintaining good reconstructed visual quality. Except for video coding, efficient methods to estimate motion include fields as divers as remote sensing, virtual reality, and content-based representation. Moreover, in the practical implement, motion estimation also can be used to GPS positioning [7] to predict the object displacement accurately and be employed to the trajectory tracking for robot manipulators [8]. The key step to the success of the aforementioned tasks is the estimation of motion.

Many fast search algorithms have been proposed to reduce the computations of BMD. The algorithms can be roughly classified into two groups, search point reduction and calculation reduction of SAD. Search point reduction involves decreasing the number of search points within a search window [9-11]. The diamond search (DS) algorithm is a typical search point reduction method [12]. The DS algorithm uses two diamond patterns and unrestricted searching steps to find the minimum distortion block. The pattern-based motion estimation algorithm has many other variations [13,14]. These algorithms indeed significantly reduce the computational complexity of motion estimation by assuming that

International Journal of Innovative Computing, Information and Control
Volume **5**, Number **9**, September **2009**
 ISSN 1349-4198
pp. 1–08-111

COARSE-TO-FINE PARTIAL DISTORTION SEARCH ALGORITHM FOR MOTION ESTIMATION

Chia-Hung Yeh, Ming-Te Wu and Shiunn-Jang Chern

Center for Wireless Multimedia Communications
Department of Electrical Engineering
National Sun Yat-Sen University
Kaohsiung 80424, Taiwan
{ yeh; chern }@mail.ee.nsysu.edu.tw

Received March 2008; revised August 2008

Abstract. *This study proposes a coarse-to-fine partial distortion search algorithm for motion estimation to accurately predict the minimum distortion region. In the proposed method, the distortion behavior is first analyzed in a search window to indicate the possible occurrence region of the global minimum distortion position. Normalized partial distortion searching is then performed on the selected region, instead of the whole search window. The proposed algorithms have a low computational complexity yet do not degrade the PSNR significantly.*
Keywords: Normalized partial distortion search, Motion estimation, Block-matching distortion, Full search, Sum of absolute difference

1. **Introduction.** Many video coding standards, such as H.26X [1,2] and MPEG-X [3,4] have been proposed for various applications [5]. These coding standards adopt motion estimation to remove the temporal redundancy of the interframes. The block-matching algorithm (BMA) is widely used to find motion vectors (MV) [6]. The most straightforward BMA is the full search (FS) algorithm, which matches all possible blocks within the restricted search area, called the search window, in the previous frame to find the block with the minimum block-matching distortion (BMD) defined as the sum of absolution difference (SAD). However, the FS algorithm requires massive computations to calculate the BMD, and is thus unsuitable for real-time implementation.

Motion estimation is the key component of video coding. Fast and accurate motion estimation method is highly desired to achieve high compression ratio while maintaining good reconstructed visual quality. Except for video coding, efficient methods to estimate motion include fields as divers as remote sensing, virtual reality, and content-based representation. Moreover, in the practical implement, motion estimation also can be used to GPS positioning [7] to predict the object displacement accurately and be employed to the trajectory tracking for robot manipulators [8]. The key step to the success of the aforementioned tasks is the estimation of motion.

Many fast search algorithms have been proposed to reduce the computations of BMD. The algorithms can be roughly classified into two groups, search point reduction and calculation reduction of SAD. Search point reduction involves decreasing the number of search points within a search window [9-11]. The diamond search (DS) algorithm is a typical search point reduction method [12]. The DS algorithm uses two diamond patterns and unrestricted searching steps to find the minimum distortion block. The pattern-based motion estimation algorithm has many other variations [13,14]. These algorithms indeed significantly reduce the computational complexity of motion estimation by assuming that the block distortion measure increases with the checking points away from the global minimum. Since the error surface within a search window produced from BMD is usually not monotonic, multiple local minimum points very often appear in the search window especially when image sequences with large motion content take place. The check points of these algorithms are, however, easily trapped into the local minima, leading to higher matching distortion than the FS algorithm.

The other group belongs to the calculation reduction of SAD. From the observation of the SAD formula, SAD is calculated by summing the absolute difference of all pixels between a macroblock (MB) of the current frame and the reference frame. Obviously, the partial SAD is always less than the total SAD between two blocks, since SAD is obtained by adding partial pixel-by-pixel differences. Bei *et al.* [15] proposed an algorithm called partial distortion search (PDS) to reduce the number of computations in vector quantization. Montrucchio *et al.* [16] proposed a fast lossless PDS algorithm that considers the distortion behavior. Cheung *et al.* [17] proposed a normalized partial distortion search (NPDS) for early rejection of impossible candidate motion vectors. NPDS normalizes the accumulated partial distortion and the current minimum distortion before comparison. Therefore, non-possible candidate MV can be rejected early in order to save computations. Cheung *et al.* [18] extended the idea of NPDS into an adjustable PDS called APDS that can adjust the prediction accuracy against search speed by a quality factor. The algorithms of this group can achieve better visual quality; however, it continues to be computation-intensive - a problem that becomes more serious with the increase of its frame size. This need inspires us to develop the proposed method jointly, considering the merits of "search points reduction" and "calculation reduction of SAD."

This study proposes a coarse-to-fine normalized partial distortion search (CFNPDS) for motion estimation. The proposed algorithm first uses a big spiral pattern to predict the possible location of the minimum distortion block. The NPDS algorithm is then performed on this selected region to find the minimum distortion block. The goal of the big spiral pattern is to reject the non-possible searching blocks early in order to speed up the motion estimation process. Experimental results show that the proposed scheme has much lower computational complexity than other algorithms such as TSS, DS and NPDS algorithms, while maintaining a similar PSNR performance to NPDS.

The rest of this paper is organized as follows. Section 2 describes the background of the NPDS algorithm. Section 3 then introduces the proposed CFNPDS method. Experimental results are shown in Section 4. Concluding remarks are given in Section 5, along with recommendations for future research.

2. **Background.**

This section briefly describes the NPDS algorithm. The FS algorithm is always computationally too complex for real-time implementation. To reduce the computational complexity, the partial distortion of the NPDS algorithm is defined, based on the halfway-stop concept, as the distortion of a group of pixels, rather than that of a single pixel. Hence, the block distortion $D(x, y; u, v)$ is split into 16 partial distortions, d_p, where every partial distortion is composed of 16 points spaced equally between adjacent points. The pth partial distortion is given by

$$d_p(x, y; u, v) = \sum_{i=0}^{3} \sum_{j=0}^{3} |I_t(x + 4i + s_p, y + 4j + t_p) - I_{t-1}(x + 4i + s_p + u, y + 4j + t_p + v)| \quad (1)$$

where (s_p, t_p) represents the offset of the upper left corner point of the pth partial distortion from the upper left corner point of the candidate block; i and j are the indices of the sub-MB at the vertical and horizontal positions, respectively, and $I_t(x, y)$ indicates the pixel intensity at coordinate (x, y) in the frame t. The term (u, v) denotes the displacement of the candidate MB. The pth accumulated partial distortion is defined as

$$D_p(x, y; u, v) = \sum_{i=1}^{p} d_i(x, y; u, v) \tag{2}$$

The NPDS matches all search points inside a search window in the same way as the FS algorithm. The search starts at the origin search point (block), and moves outward with a spiral scanning path. During each block matching, the NPDS compares each accumulated partial distortion D_p with the normalized minimum distortion NMD, which is defined as

$$NMD = \frac{p}{16} D_{\min} \tag{3}$$

where $D_{\min}$ indicates the current minimum distortion. The process begins from $p = 1$, and proceeds toward $p = 16$. The comparison is stopped if the accumulated partial distortion of the candidate vectors is larger than the current NMD, which is defined as

$$D_p > \frac{p}{16} D_{\min} \tag{4}$$

3. Proposed Coarse-to-Fine Normalized Partial Distortion Search Algorithm. This section describes the proposed CFNPDS algorithm. Although the NPDS algorithm increases the probability of early rejection for impossible candidate MV, it tests all search points in a search window with (4). However, not all search points need to be tested if the occurrence area of the minimum distortion can be accurately predicted. Only testing the search points that need to be tested significantly decreases the number of stages in calculating the accumulated partial distortion calculation. In a search window, the BMD can be modeled as a parabolic surface [10]. However, the real surface is not flat, causing problems for the pattern-based methods in detecting its global minimum point. To reduce the chance of finding local minimum points rather than the global minimum, the proposed method first predicts the region where the real minimum point is most likely to occur, and later performs the NPDS algorithm in relation to this area.

As mentioned above, the first step in the proposed algorithm is to predict the possible location of the global minimum distortion point using a big spiral pattern composed of nine search points, as shown in Figure 1. The nine points are uniformly selected from the search window. Each point represents a partial region of the whole search window. The region with the smallest BMD point has the highest probability of occurrence of the global minimum distortion point. To do so, we only need to compare the area points of the global minimum distortion instead of the whole search points.

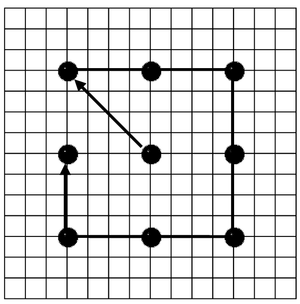

FIGURE 1. The 9-point big spiral pattern

The SAD of the center candidate in a search window is first calculated, and the distortion of the center point is assigned as the current minimum distortion $D_{\min}$. The NPDS algorithm is then employed to compare the other eight points in the big spiral pattern and the current minimum distortion $D_{\min}$ by applying (4) step by step to obtain the minimum point among these nine points. This approach reduces the computational complexity of the first step by rejecting the impossible points early, instead of performing full SAD calculation. The representative point (u_r, v_r) with the smallest distortion value among the nine points is then obtained. This point indicates the possible occurrence region of the global minimum distortion point, defined as

$$(u_r, v_r) = \arg(\min \{D_P(u_{BP}, v_{BP}) | u_{BP}, v_{BP} \in \{-4, 0, 4\}\}) \quad (5)$$

where (u_{BP}, v_{BP}) is the displacement of the big spiral pattern. The occurrence position of the minimum distortion region prediction has two possible cases. The first case is the minimum distortion value of the nine points located at the central point. Figure 2 shows the selected region, which consists of 9×9 pixels, and is denoted as

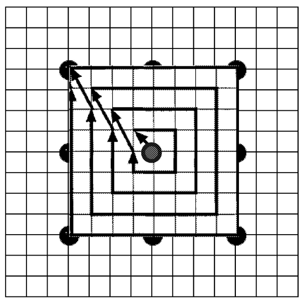

FIGURE 2. 9×9-pixel region with the minimum value at the center point

$$R_{center}(u, v) = \{(u, v) | u_r - 4 \le u \le u_r + 4, v_r - 4 \le v \le v_r + 4\} \quad \text{if } (u_r, v_r) = (0, 0) \quad (6)$$

The NPDS algorithm is then performed on this selected region to obtain the optimal MV. The 9×9 pixels region is considered because it is not exceeding the eight points of the outlier rim, needs less computation and maintains good PSNR performance.

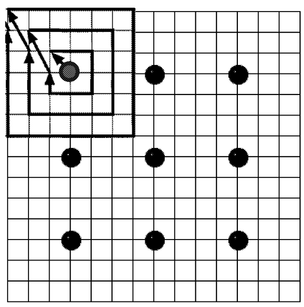

FIGURE 3. 7×7-pixel region when minimum value occurred at the outlying rim

In the second case, the minimum distortion value of the nine points is located at the outlying rim of the spiral pattern. Figure 3 shows the selected region, which consists of 7×7 pixels, and which is denoted as

$$R_{rim}(u, v) = \{(u, v) | u_r - 3 \le u \le u_r + 3, v_r - 3 \le v \le v_r + 3\} \quad \text{if } (u_r, v_r) \ne (0, 0) \quad (7)$$

Similarly, the NPDS algorithm is performed on this selected region to yield the optimal MV. The consideration reason of 7×7 pixels region is the same as the reason of the first case. In summary, the proposed CFNPDS algorithm performs the following steps to obtain the MV. Figure 4 shows a flowchart of the proposed CFNPDS algorithm.

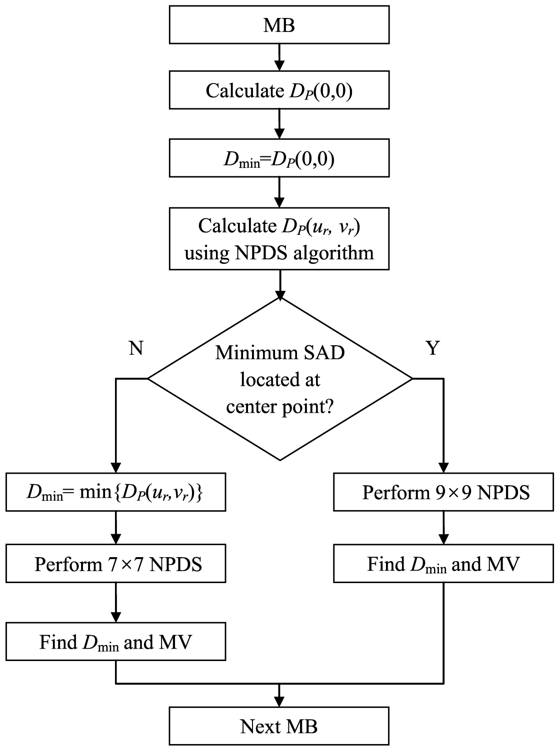

FIGURE 4. Flowchart of the proposed CFNPDS algorithm

1) Calculate the SAD of the central point in a search window for input MB, and take this as the initial minimum distortion value.
2) Compute the other eight points in the big spiral pattern according to (4) to find the point with the minimum distortion value. If the minimum value locates at the center position, then go to the next step. Otherwise, go to step 4).
3) Perform the NPDS algorithm in this 9×9-pixel region of the center point according to (6) to obtain the optimal MV.
4) Run the NPDS algorithm in the 7×7-pixel region with the minimum distortion value at the outlying rim of the nine points as its center according to (7) to obtain the optimal MV.

Through Figure 4, we can see that the proposed method shrinks the search region compared to the NPDS scheme and reduces massive computations.

4. **Experimental Results.** The proposed CFNPDS algorithm was simulated on several well-known CIF (352×288) video sequences, namely *Missa*, *Stefan, Garden* and *TableTennis*, which have 149, 250, 86 and 250 frames, respectively. To evaluate the performance of the proposed scheme, the PSNR performance and computations of FS, TSS, DS, NPDS and the proposed CFNPDS algorithms were compared. The search range was set to ± 7 pixels.

Tables 1 and 2 list the PSNR comparisons and computational complexity of the FS, TSS, DS, NPDS and CFNPDS algorithms. As demonstrated in Table 1, the proposed CFNPDS algorithm showed better PSNR than the TSS and DS algorithms. The average PSNR degradation resulting from the CFNPDS algorithm was only 0.2719dB compared

Table 1. Average PSNR for different schemes with respect to different test sequences (Search window size: ± 7 pixels)

	FS	TSS	DS	NPDS	CFNPDS
Missa (149frames)	39.0889	38.4585	38.7414	38.9616	38.7540
Stefan (250frames)	23.9801	22.8105	23.1926	23.8136	23.3918
Garden (86frames)	23.9653	23.3912	23.7730	23.8384	23.8024
TableTennis (250frames)	31.2653	30.5454	30.6404	31.0665	30.6440

to the NPDS algorithm. The maximum degradation of CFNPDS was 0.4225dB for the *TableTennis* sequence. However, the average PSNR performance resulted by the CFN-PDS algorithm is 0.0612dB better than DS scheme and the maximum outperformance of CFNPDS is 0.1992dB for the *Stefan* sequence. The proposed CFNPDS algorithm is the average 0.346dB better than TSS method and the maximum outperformance of CFNPDS is 0.5813dB for the *Stefan* sequence. In a word, the proposed scheme has a better PSNR algorithm than the TSS and DS algorithms, and a smaller PNSR degradation than the NPDS algorithm.

Table 2 shows the computational complexity comparisons among the FS, TSS, DS, NPDS and CFNPDS algorithms. The terms *Abs.*, *Add.*, *Com.*, and *LS.* indicate the absolute, addition/subtraction, comparison, and left-shift operations, respectively. As revealed in Table 2, the proposed algorithm was at least 23.90 times faster than the FS algorithm on average. The maximum speed-up ratio of the proposed algorithm was about 27.38 times that of the FS algorithm for the *TableTennis* sequence. The proposed algorithm is on average 2.15 times faster than the DS algorithm, on average 2.96 times faster than the TSS scheme, and has the better PSNR performance. Moreover, the proposed algorithm was on average 2.23 times faster than the NPDS algorithm with the slight PSNR degradation.

Table 3 shows the hit rates of the optimal MV occurring at the predicted region based on a big spiral pattern. The hit rates of the *Stefan, Garden* and *TableTennis* sequences were approximately 98%, as demonstrated in Table 3. The proposed big spiral pattern can efficiently predict the occurrence region of the optimal MV, thus significantly reducing the computation complexity can be while maintaining good PNSR performance. The hit rate of the *Missa* sequence is lower than 95%. This rate is owing to the homogenous background of the *Missa* sequence, leading to multiple local minimum points. Therefore, the global minimum region of this sequence is difficult to detect by the proposed spiral pattern. However, the PSNR degradation is slight due to the small average BMD in this context [19].

5. **Conclusions.** This study proposes a CFNPDS algorithm to improve the coding performance and speed of the motion estimation process in video coding. By observing the distortion behavior in a search window, a big spiral pattern is employed to predict the possible occurrence region of the best minimum distortion position, and later perform the NPDS algorithm on this area. According to our simulation results, the proposed algorithm provides better PSNR performance and coding speed than the conventional algorithms such as the DS algorithm. Meanwhile, the proposed scheme is faster than the NPDS algorithm, while having slight PNSR degradation. As we know, homogeneous background will lead to multiple local minimum points, degrading visual quality. The SAD value of the center point of the search window, or other points in the proposed big spiral pattern, is a good indicator that distinguishes whether the current block is in

Table 2. Average number of operations per block for different schemes with respect to different test sequences (Search window size: ±7 pixels)

(a) *Missa*

BMA	Abs.	Add.	Com.	LS.	Total	Speedup
FS	51945.42	103890.84	426.40	0	156262.67	1
TSS	5952.01	11904.03	65.96	0	17920.01	8.72
DS	4242.45	8484.90	60.76	0	12788.11	12.22
NPDS	4669.77	9339.54	515.35	200.54	14725.21	10.61
CFNPDS	2070.67	4141.34	260.25	65.92	6538.19	23.90

(b) *Stefan*

BMA	Abs.	Add.	Com.	LS.	Total	Speedup
FS	52087.22	104174.44	427.57	0	156689.22	1
TSS	5949.23	11898.46	58.10	0	17905.79	8.75
DS	4812.15	9624.30	76.48	0	14512.93	10.79
NPDS	4375.16	8750.32	497.55	200.18	13823.21	11.33
CFNPDS	1869.41	3738.82	178.14	68.39	5854.76	26.76

(c) *Garden*

BMA	Abs.	Add.	Com.	LS.	Total	Speedup
FS	51688.31	103376.61	424.29	0	155489.21	1
TSS	5920.94	11841.88	57.72	0	17820.54	8.72
DS	4374.57	8749.15	60.59	0	13184.31	11.79
NPDS	4191.42	8382.85	484.35	198.66	13257.28	11.73
CFNPDS	1948.15	3896.31	192.61	68.11	6105.18	25.46

(d) *TableTennis*

BMA	Abs.	Add.	Com.	LS.	Total	Speedup
FS	52087.22	104174.44	427.57	0	156689.22	1
TSS	5944.61	11889.21	58.08	0	17891.90	8.76
DS	3775.63	7551.25	57.71	0	11384.59	13.76
NPDS	3860.62	7721.24	465.39	201.37	12248.61	12.79
CFNPDS	1820.00	3640.00	185.49	76.25	5721.73	27.38

Table 3. Hit rate in the predicted region with respect to different test sequences

Sequences	*Missa*	*Stefan*	*Garden*	*TableTennis*
Hit rate (%)	94.89	97.87	97.90	98.58

homogeneous background. According to this perspective, we will improve the proposed algorithm so as to overcome the multiple local minimum points in the future.

Acknowledgment. The authors would like to thank the National Science Council, Taiwan for financially supporting this research under Contracts No.NSC 96-2628-E-110-020-MY2.

REFERENCES

[1] CCITT SGXV, Description of reference model 8 (RM8), Document 525, Working Party XV/4, Specialists Group on Coding for Visual Telephony, 1989.

[2] ITU Telecommunication Standardization Sector LBC-95, Study Group 15, Working Party 15/1, Expert's Group on Very Low Bitrate Visual Telephony, from Digital Video Coding Group, Telenor Research and Development, or via URL:http://www.nta.no/brukere/DVC/tmn5, 1998.

[3] ISO/IEC CD 11172-2 (MPEG-1 Video), Information technology—Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits/s: Video, 1993.

[4] ISO/IEC CD 13818-2–ITU-T H.262 (MPEG-2 Video), Information technology—Generic coding of moving pictures and associated audio information: Video, 1995.

[5] H. Yu, F. Pan and Z. Lin, Content adaptive rate control for H.264, *International Journal of Innovative Computing, Information and Control*, vol.1, no.4, pp.685-700, 2005.

[6] C. Stiller and J. Konrad, Estimating motion in image sequences, *IEEE Signal Process. Mag.*, vol.16, no.4, pp.70-91, 1999.

[7] Y. Kubo, S. Fujita and S. Sugimoto, Estimation and validation of integer ambiguity in carrier phase GPS positioning, *International Journal of Innovative Computing, Information and Control*, vol.4, no.2, pp.153-164, 2008.

[8] K. Najim, E. Ikonen and E. Gomez-Ramirez, Trajectory tracking control based on a genealogical Decision tree controller for robot manipulators, *International Journal of Innovative Computing, Information and Control*, vol.4, no.1, pp.53-62, 2008.

[9] S. Kappagantula and K. R. Rao, Motion compensated interframes image prediction, *IEEE Trans. on Commun.*, vol.33, no.9, pp.1011-1015, 1985.

[10] C.-J. Kuo, C.-H. Yeh and S. F. Odeh, Polynomial search algorithm for motion estimation, *IEEE Trans. Circuits Syst. Video Technol.*, vol.10, no.5, pp.813-818, 2000.

[11] C.-H. Yeh, M.-T. Wu and S.-J. Chern, Correlation-based normalized partial distortion search algorithm for motion estimation, *Optical Engineering*, vol.47, no.7, pp.077003, 2008.

[12] S. Zhu and K.-K. Ma, A new diamond search algorithm for fast block-matching motion estimation, *IEEE Trans. Image Process.*, vol.9, no.2, pp.287-290, 2000.

[13] T. Koga, K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, Motion-compensated interframe coding for video conferencing, *Proc. NTC81*, G5.3.1-G5.3.5, 1981.

[14] J.-Y. Tham, S. Ranganath, M. Ranganath and A. A. Kassim, A novel unrestricted center-biased diamond search algorithm for block motion estimation, *IEEE Trans. Circuits Syst. Video Technol.*, vol.8, no.4, pp.369-377, 1998.

[15] C.-D. Bei and R.-M. Gray, An improvement of the minimum distortion encoding algorithm for vector quantization, *IEEE Trans. Commun.*, vol.33, no.10, pp.1132-1133, 1985.

[16] B. Montrucchio and D. Quaglia, New sorting-based lossless motion estimation algorithms and a partial distortion elimination performance analysis, *IEEE Trans. Circuits Syst. Video Technol.*, vol.15, no.2, pp.210-220, 2005.

[17] C.-K. Cheung and L.-M. Po, Normalized partial distortion search algorithm for block motion estimation, *IEEE Trans. Circuits Syst. Video Technol.*, vol.10, no.3, pp.417-422, 2000.

[18] C.-K. Cheung and L.-M. Po, Adjustable partial distortion search algorithm for fast block motion estimation, *IEEE Trans. Circuits Syst. Video Technol.*, vol.13, no.1, pp.100-110, 2003.

[19] Y. Nie and K.-K. Ma, Adaptive rood pattern search for fast block-matching motion estimation, *IEEE Trans. Image Process.*, vol.11, no.12, pp.1442-1449, 2002.