

# Generalized Partial Distortion Search Algorithm for Fast Block Motion Estimation

Chun-Ho Cheung and Lai-Man Po

Email: chcheung@image.cityu.edu.hk and eelmpo@cityu.edu.hk

Department of Electronic Engineering, City University of Hong Kong.

83 Tat Chee Avenue, Kowloon, Hong Kong.

**Abstract**—The quality against speed control for real-time video applications, such as the speed-oriented video conferencing or the high quality video entertainment, usually absents from many traditional fast block motion estimators. In this paper, a novel block-matching algorithm for fast motion estimation named generalized partial distortion search algorithm (GPDS) is proposed. It uses halfway-stop technique with progressive partial distortion (PPD) to increase the chance of early rejection of impossible candidate motion vectors at very early stages. Simulations on PPD show that 28 to 38 times computational reduction with only 0.45-0.50dB PSNR performance degradation as compared to full search algorithm. In addition, a new normalized partial distortion comparison method is also proposed for enabling the control of searching speed against the prediction quality by a speedup factor  $k$ . This method also generalizes the conventional partial distortion search algorithm when  $k$  is equal to 1, and the normalized partial distortion search algorithm (NPDS) when  $k$  is equal to infinity. Experimental results show that GPDS with use of PPD could provide PSNR performance very close to full search algorithm and NPDS with 7 to 17 times and 22 to 33 times speedup, respectively, as compared to full search algorithm.

**Keywords**— Generalized partial distortion search algorithm, quality control, speedup factor, motion estimation.

## I. INTRODUCTION

The major compression gain in many video coding like ISO MPEG-1/2/4 and ITU-T H.261/263 is achieved by motion estimation. Motion estimation efficiently removes the temporal redundancy between successive frames by block-matching algorithms (BMA). Block-based motion estimation is the most practical approach to obtain motion compensated prediction frames. It divides frames into equal-sized rectangular blocks and finds out the displacement of the best-matched block from previous frame as the motion vector to the block in the current frame within a search region. Full search (FS) algorithm is the most straightforward brute force BMA, which provides the optimal results by matching all possible candidates within a search window ( $\pm w$  pixels). In contrast, it is the most computational intensive as compared to other popular traditional fast BMA like three-step search (3SS) [1], new three-step search (N3SS) [2] and four-step search (4SS) [3], which reduce computational complexity by limiting the number of search points within the search window. However, traditional BMA are easily trapped into local minimum points resulting in loss of visual quality as the mean absolute distortion (MAD) is always higher than FS. Recently, some fast search algorithms perform block matching as in FS without limitation of checking points, especially the normalized partial distortion search algorithm (NPDS) [4]. NPDS introduces the normalized cumulative partial distortion criteria for early rejection of impossible candidate motion vectors (CMV). However, they all lack the control between speed and quality. In this paper, a novel fast BMA named generalized partial distortion search algorithm (GPDS) is proposed. The proposed algorithm consists of two parts. The first part is to increase the speed of NPDS by introduction of progressive par-

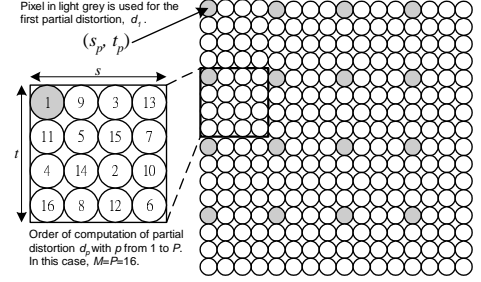


Fig. 1. Computation order of partial distortion  $d_p$ ,  $p = 1$  to 16, of  $16 \times 16$  block.  $(s_p, t_p)$  is the offset of the upper left corner of a partial distortion from the upper left corner of the candidate block

tial distortion (PPD) at very early stages. The second part is a new normalized partial distortion comparison method with speedup control factor against different quality. Experimental results of GPDS with PPD are also given to show the trade-off between searching speed and the prediction quality for different applications.

## II. PROGRESSIVE PARTIAL DISTORTION

Suppose the block size is  $M \times M$ . Let  $a_{x,y}$  and  $b_{x,y}$  be the pixel value of row  $x$  and column  $y$  of block  $a$  and  $b$ , in current and previous frame, with horizontal and vertical sampling by  $q$  and  $r$  pixels, respectively. The block distortion measure (BDM)  $d_{(x,y;u,v;q,r;M,M)}$  of  $b_{x+u,y+v}$ , with  $(u,v)$  displacement from  $a_{x,y}$  is,  $q=r=1$ , using sum absolute error (SAE):

$$d_{(x,y;u,v;q,r;M,M)} = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} |a_{x+qi,y+rj} - b_{x+u+qi,y+v+rj}| \quad (1)$$

Conventional partial distortion search algorithm (PDS) [5] provides optimal result equal to FS with speedup ratio twice or more than that of FS. PDS rejects impossible CMV by means of half-way stop technique with pixel-by-pixel based partial distortion comparison to the current minimum distortion. To reduce the number of comparison and increase the chance of early rejection of impossible CMV, NPDS divides the matching blocks into  $P$  groups ( $P=16$ ), as shown in Fig.1, and compares each group of  $M^2/P$  pixels based partial distortion with the normalized current minimum distortion,  $D_{MIN}$ . Experimental results in [4] show that NPDS outperforms other BMA such as 3SS with PSNR performance close to FS and highly reduces the computations 12-13 times by early rejection. It results in a saving of multiples of 16 pixels-operations for each impossible CMV. However, NPDS limits the maximum computational reduction to  $\frac{M^2}{(\text{no. of pixels at the first partial distortion})}$  or  $\frac{M^2}{M^2/P}=16$  times for  $M=P=16$ . In order to further increase the rejection rate more efficiently, we propose to use progressive partial distortion (PPD) at the first few stages of the NPDS, such as the

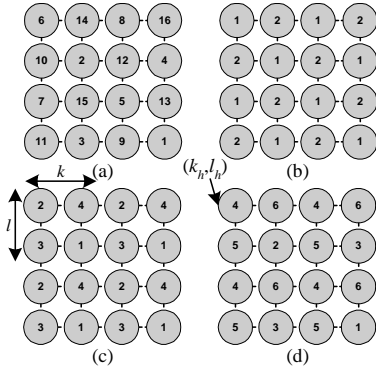


Fig. 2. Different regular subsampling patterns on the first partial distortion. Computation order of group of pixels is shown in the label of a pixel.  $(k_h, l_h)$  is the offset of upper left corner of the first PPD from the upper left corner of the first partial distortion.

first partial distortion, which is further divided into  $H$  smaller dimension and equal-sized partial distortions. It thus increases the maximum computational reduction of NPDS from 16 times to  $\frac{M^2}{M^2/(PH)}=64$  times if  $H=4$ .

With PPD, matching blocks are firstly divided into  $P$  equal-sized partial distortions in which the dimension of the first partial distortion  $d_1$  (i.e.  $p=1$ ) is further reduced from  $\frac{M}{s} \times \frac{M}{t}$  to  $\frac{M}{sk} \times \frac{M}{tl}$  in order to support  $H$  equal-sized progressive partial distortions. Each partial distortion  $d_p|_{1 \leq p \leq P}$  is firstly sampled by horizontal  $s$  and vertical  $t$  pixels, respectively. The first partial distortion  $d_1$  is subsampled by horizontal  $k$  and vertical  $l$  pixels, respectively. Thus,  $d_1$  is replaced by  $H$   $d_{p,h}|_{p=1, 1 \leq h \leq H}$  and the two types of partial distortions are defined as in Eqn(2) and Eqn(3), respectively.

$$d_{p,h} = d_{(x+s_p+k_h, y+t_p+l_h; u, v; k, l; \frac{M}{sk}, \frac{M}{tl})} |_{p=1, 1 \leq h \leq H} \quad (2)$$

$$d_p = d_{(x+s_p, y+t_p; u, v; s, t; \frac{M}{s}, \frac{M}{t})} |_{2 \leq p \leq P} \quad (3)$$

The total number of partial distortions are  $G=H+P-1$ . The original  $p$ -th cumulative partial distortion  $D_p$  of NPDS and its normalization  $D_{NORM(p)}$  cumulated to  $n$  pixels are defined as in Eqn(4).

$$D_p = \sum_{k=1}^p d_k; \quad D_{NORM(p)} = \frac{M^2}{n} D_p \quad (4)$$

where  $n = p \times M^2/P$  for  $1 \leq p \leq P$

They are rewritten as in Eqn(5) for the  $g$ -th cumulative partial distortion  $D'_g$  and Eqn(6) for its normalization  $D'_{NORM(g)}$  cumulated to  $n$  pixels after PPD is introduced.

$$D'_g = \sum_{k=1}^g d'_k; \quad (5)$$

$$\text{where } d'_g = \begin{cases} d_{p,h} & \text{for } p=1, 1 \leq g \leq H \\ d_p & \text{for } 2 \leq p \leq P, (H+1) \leq g \leq G \end{cases}$$

$$D'_{NORM(g)} = \frac{M^2}{n} D'_g; \quad (6)$$

$$\text{where } n = \begin{cases} h \times \frac{M^2}{PH} & \text{for } p=1, 1 \leq g \leq H \\ p \times \frac{M^2}{P} & \text{for } 2 \leq p \leq P, (H+1) \leq g \leq G \end{cases}$$

BMA	LS	ABS	Comp.	+/-	Total	SpeedUp	PSNR
FS	—	52870	105534	206	158610	1	26.92
PDS	—	22838	45469	1617	69924	2.27	26.92
3SS	—	4022	8021	258	12301	12.89	25.43
4SS	—	3228	6437	204	9869	16.07	26.18
N3SS	—	3795	7569	241	11605	13.67	25.98
NPDS	55	4234	8292	454	13035	12.17	26.69
PPDS(v1)	258	1071	2060	628	4017	39.48	25.95
PPDS(v2)	133	1313	2480	529	4455	35.60	26.21
PPDS(v3)	81	1759	3350	482	5672	27.96	26.42
PPDS(v4)	62	2614	5050	462	8188	19.37	26.59
PPDS(v5)	158	1148	2120	561	3987	39.78	26.06

TABLE I

PSNR PERFORMANCE COMPARISON WITH DIFFERENT PPD AGAINST COMPUTATION REDUCTION ON SEQUENCE "TENNIS". [REMARK: PPDS.v1=1-PIXEL, v2=2-PIXEL, v3=4-PIXEL (PROPOSED), v4=8-PIXEL GROUPING AND PPDS.v5=(1,1,2,4,4,4) SAMPLE SEQUENCE.]

BMA	LS	ABS	Comp.	+/-	Total	SpeedUp	PSNR
FS	—	52870	105534	206	158610	1	26.17
PDS	—	16868	33529	1244	51641	3.07	26.17
3SS	—	2788	5552	181	8521	18.61	25.66
4SS	—	2211	4405	138	6754	23.48	25.82
N3SS	—	2347	4676	148	7171	22.12	25.93
NPDS	20	3782	7369	426	11597	13.68	25.97
PPDS(v1)	116	692	1220	514	2542	62.39	25.35
PPDS(v2)	53	898	1610	459	3020	52.52	25.55
PPDS(v3)	28	1309	2420	435	4192	37.84	25.72
PPDS(v4)	21	2142	4090	428	6681	23.74	25.87
PPDS(v5)	86	728	1260	494	2568	61.76	25.45

TABLE II

PSNR PERFORMANCE COMPARISON WITH DIFFERENT PPD AGAINST COMPUTATION REDUCTION ON SEQUENCE "FOOTBALL". [REMARK: PPDS.v1=1-PIXEL, v2=2-PIXEL, v3=4-PIXEL (PROPOSED), v4=8-PIXEL GROUPING AND PPDS.v5=(1,1,2,4,4,4) SAMPLE SEQUENCE.]

Fig. 2 shows different subsampling patterns on the first partial distortion. Regular subsampling  $\psi \{ \psi \in \Psi; \Phi = \Psi \cup \Delta \}$  with  $i$  pixel(s) taken out at a time from a group of  $S=M^2/P$  pixels has combinations of  $\sum_{i=1}^S \left[ \prod_{j=0}^{\lfloor S/i \rfloor - 1} (S - ij C_i) \right]$ , where  $nCr = \frac{n!}{(n-r)!r!}$ . Thus, the total combination  $\Phi$  if including the remaining progressive regular and irregular subsampling  $\Delta$  is very large. Regularity favors practical implementations. Fig.2(c) is the PPD (version 3) proposed for progressive partial distortion search algorithm (PPDS) by employing total 19 partial distortions:  $H=4$  equal-sized PPD  $d_{1,h}$ , and 15 partial distortions  $d_p|_{2 \leq p \leq 16}$  ( $P=16$ ). With  $M=P=16$ ,  $s=t=4$ ,  $H=4$  and  $k=l=2$ ,  $D'_{NORM(g)}|_{1 \leq g \leq H}$  and  $D'_{NORM(g)}|_{(H+1) \leq g \leq G}$  are equal to  $D'_{NORM(h)}$  and  $D_{NORM(p)}$  respectively, such that  $D'_{NORM(H)} = D_{NORM(1)}$  as shown in Eqn(7).

$$D'_{NORM(g)} = \begin{cases} D'_{NORM(h)} = \frac{64}{h} D'_h & \text{for } p=1; 1 \leq g, h \leq H \\ D_{NORM(p)} = \frac{16}{p} D_p & \text{for } 2 \leq p \leq P; (H+1) \leq g \leq G \end{cases} \quad (7)$$

PPDS performs normalized cumulative partial distortion criteria  $D'_{NORM(g)} > D_{MIN}$  and rejects any impossible CMV if  $D'_{NORM(g)}$  is greater than the current minimum distortion  $D_{MIN} = \min_{(x,y) \in W} D_G$ , where  $W=[x \pm w, y \pm w]$ . Other-

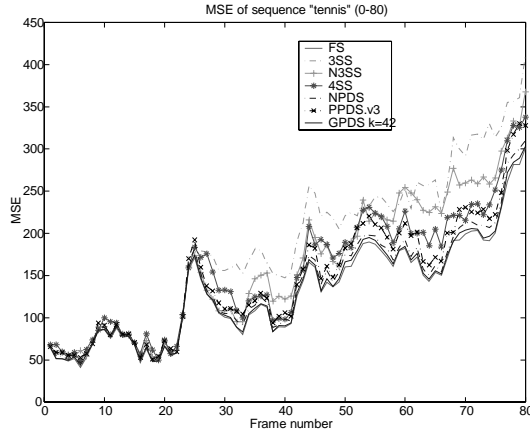


Fig. 3. MSE performance comparison on “tennis”.

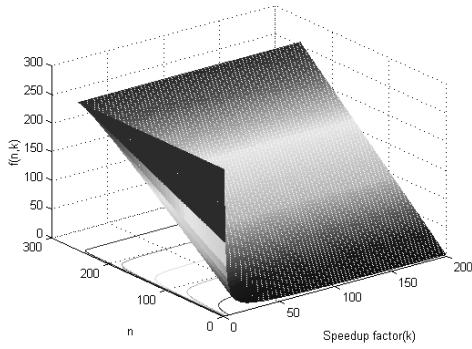


Fig. 4. Generalization function  $f(n, k)$  for GPDS comparison.

wise,  $D'_{NORM(g)}$  is recomputed with inclusion of next partial distortion  $d'_{g+1}$  and repeats the comparison until the last partial distortion  $G$ . Different kinds of PPD are studied and the performances on sequences “tennis” and “football” are shown in Table I and II. PPD with the lowest regular dimension, such as grouping of 1 pixel at a time in Fig.2(a), provides highest computational reduction due to the highest very early rejection rate  $\frac{D'_h}{h} > \frac{D_{MLN}}{256}$ . However, it gives the worse prediction quality among PPD. PPD with the highest regular dimension, such as grouping of 8 pixels in Fig.2(b), it gives the lowest computational reduction but with the best prediction quality. The lower the dimension of grouping is, the larger the computation reduction is. PPDS as in Fig.2(d) shows the intermediate results on both prediction quality and speedup ratio with lower to higher dimensionality, i.e. (1,1,2,4,4,4) sample sequences, in which different levels of sub-grouping perform at the first partial distortion. Table I and II show that PPDS provides speedup ratio from 19.37 to 62.39 times computational reduction with 0.30-0.97dB degradation on average PSNR performance as compared to FS. The proposed PPDS (v3: grouping of 4 pixels) with sub-sampling pattern shown in Fig. 2(c) provides more than twice of speedup ratio of 3SS and with better PSNR performance by less than 1dB. As compared to NPDS, it provides 2.30-2.77 times speedup ratio of NPDS with only less than 0.27dB degradation on PSNR performance. In general, PPDS(v3) results in 28-38 times speedup with 0.45-0.50dB degradation from FS. The proposed PPDS on sequence “tennis”, as shown in Fig.3, gives MSE performance very close to or even better than traditional BMA

such as 3SS, N3SS and 4SS.

### III. GENERALIZED PARTIAL DISTORTION COMPARISON

GPDS is to provide generalization of the normalized cumulative partial distortion  $\tilde{D}_{NORM}$  so that the NPDS matching criteria against  $D_{MLN}$  can be adjusted to provide different speedup ratio and prediction quality. The generalized normalization of  $n$ -pixel cumulative partial distortion  $D_n$  from Eqn(4) can be rewritten as:

$$\tilde{D}_{NORM(n)} = \frac{M^2}{f(n)} D_n; \text{ where } f(n)=n. \quad (8)$$

Since the proposed PPDS(v3) at the first partial distortion of NPDS gives  $H$  equal-sized PPD with lower dimension, it increases the early rejection rate for the first group of  $M^2/P=16$  pixels to  $H=4$  times. It lowers the computations and thus increases the speedup ratio by about three times of NPDS but with MSE performance degraded slightly. Generalization function  $f(n, k)$  is designed to replace  $f(n)$  by introducing a quality factor or speedup factor  $k$ , which relates the PSNR performance from conventional PDS to NPDS. The generalized  $\tilde{D}_{NORM(n)}$  provides matching criteria for optimal result as in PDS when  $k=1$ , i.e.  $f(n, k)|_{k=1}=M^2$ . When  $k \rightarrow \infty$ , GPDS gives the performance as in NPDS, i.e.  $f(n, k)|_{k \rightarrow \infty}=n$ . Thus,  $f(n, k)$  is defined as in Eqn(9) and plotted as in Fig.4.

$$f(n, k) = n + \left( \frac{M^2 - n}{k} \right) \quad (9)$$

In general,  $f(n, k)$  starts at  $M^2$  (i.e. PDS matching criteria) and going to become  $f(n)=n$  (i.e. NPDS matching criteria) as  $k$  increases from 1 to infinity. Thus, it is suitable for adjustment of speedup versus quality between PDS and NPDS. As  $f(n, k)$  decreases with  $k$  dramatically for small  $n$ , but gets to become  $M^2$  for larger  $n$ . It implies  $\tilde{D}_{NORM(n)}$  increases with  $k$  dramatically at small  $n$ , and thus favors early rejection purposes. With the dramatical slope of  $f(n, k)$ , NPDS favors to give result near to that of PDS (at small value of  $k$ ) by rejection of impossible CMV at early stages (small value of  $n$ ).

### IV. EXPERIMENTAL RESULTS

The proposed algorithm GPDS with PPD is simulated using the luminance of the popular SIF (360×240) video sequence “tennis” and “football”. The block size and search window used are  $16 \times 16$  ( $M \times M$ ) and  $\pm 7$  ( $w$ ), respectively. In NPDS, each block is sampled evenly in both horizontal and vertically by  $s=4$  and  $t=4$  pixels, respectively, into  $P=16$  partitions. For PPDS(v3) and GPDS, the first partial distortion  $d_1$  is subsampled by  $k=2$  and  $l=2$  into  $H=4$  PPD. All 3SS, 4SS and N3SS are implemented to use halfway-stop BDM for fair comparison. Their predication quality in terms of PSNR and MSE, and computational complexity are compared.

Results using various PPD can be referred to section II, Table I and II. In general, PPDS gives 19.37 to 62.39 times speedup with 0.30 to 0.97dB PSNR degradation compared to FS. The PPDS (v3) employs total  $G=19$  partial distortions with compromised performance and is generalized by  $f(n, k)$  to form GPDS for normalized partial distortion comparison.

Table III compares the speedup ratio and PSNR performance of different BMA against GPDS of different speedup factor  $k$ . GPDS maintains its PSNR performance very close to FS at 7.24 and 17.21 times speedup, for sequence “tennis” and “football”, respectively. As compared to traditional BMA, firstly, GPDS has a speedup ratio of 1.49-2.11 times with better or similar

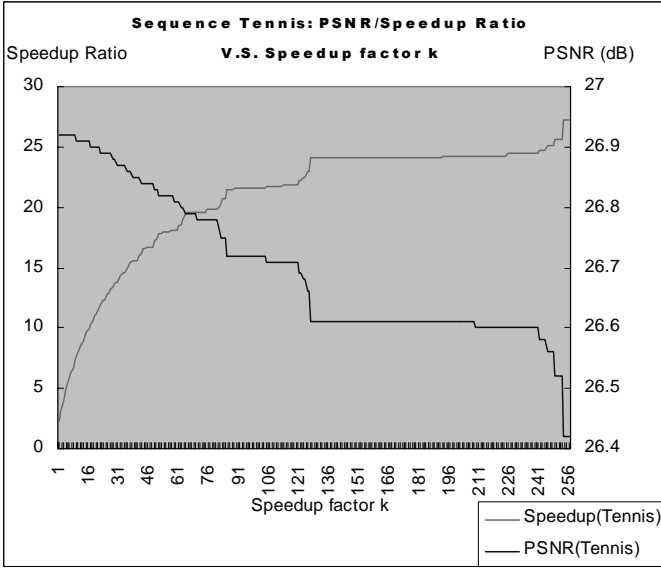


Fig. 5. PSNR performance and speedup ratio against speedup factor  $k$  on sequence "Tennis".

PSNR performance. Secondly, it also gives better PSNR performance by 0.21-1.41dB with just faster searching speed at 16.24( $k=42$ ) times for "tennis" and 23.56( $k=31$ ) times for "football", respectively. As compared to NPDS, GPDS has PSNR performance very close to NPDS at  $k=122$  on both sequences with computational reduction of 1.82-2.37 times. Fig.3 shows MSE performance of various BMA on "tennis" in which GPDS at  $k=42$  almost overlaps the MSE performance of FS at 16.24 times speedup, which is just faster than all traditional BMA.

It is noted that multiplication operations exist in Eqn(8). In PPDS comparison, all multiplication operations are translated into combinations of "left-shift" and "addition" operations since the numerator  $M^2$  of the matching criteria can be simplified into power of 2 with the values of  $H$  and  $P$ , which are also power of 2 in  $M^2 D_n > f(n) D_{MIN}$ . However, for generalization function in Eqn(9), smooth curves of speedup ratio and distortion against speedup factor  $k$  are expected theoretically from PDS to NPDS, but impractical for implementation. Thus, nineteen  $f(n, k)$  values are computed for desired  $k$  in advance by means of integer precisions and round-off operations from  $k=1$  to 256 such that  $f(n, k)$  saturates at  $k=256$ . It results in stepping-behavior of the speedup ratio and PSNR performance against  $k$  as shown in Fig.5 and Fig.6. Since dramatical behavior exists in the small  $k$  and the proposed working value for  $k \leq 100$ , which provides obvious speedup ratio up to 21.60-32.04 times with 0.18-0.20dB slightly degradation compared to FS.

## V. CONCLUSIONS

In this paper, progressive partial distortion (PPD) is proposed at the first few stages of NPDS to increase the early rejection rate of impossible candidate motion vectors. Simulation on NPDS with different PPD provides speedup ratio from 19.37 to 62.39 times computational reduction with 0.30-0.97dB degradation on PSNR performance as compared to FS. In addition, generalized partial distortion search algorithm (GPDS) is proposed. It introduces a speedup factor  $k$  to the normalized partial distortion comparison so that NPDS can be adjusted to provide an optimization on the tradeoff between speed and quality from PDS to NPDS. Experimental results show that the

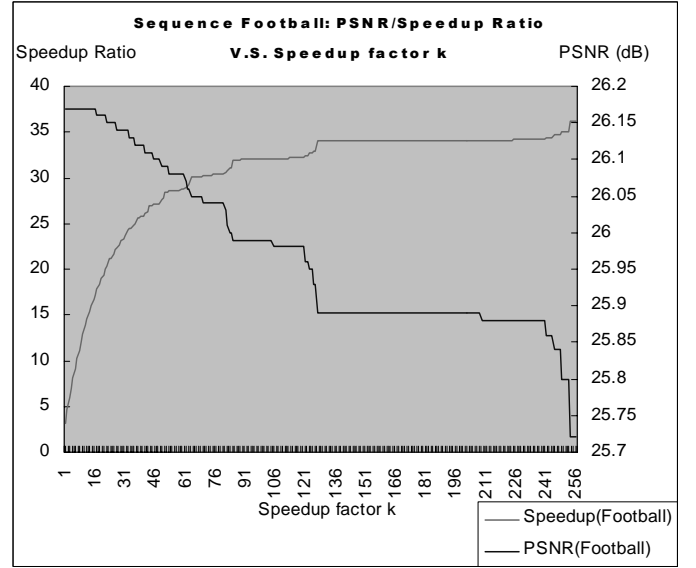


Fig. 6. PSNR performance and speedup ratio against speedup factor  $k$  on sequence "Football".

Sequence "tennis"			Sequence "football"		
BMA(k)	SpUp	PSNR	BMA(k)	SpUp	PSNR
GPDS(9)	7.24	26.92[ $\sim$ FS]	GPDS(16)	17.21	26.17[ $\sim$ FS]
GPDS(42)	16.24	26.84	GPDS(31)	23.56	26.14
GPDS(100)	21.60	26.72	GPDS(100)	32.04	25.99
GPDS(122)	22.19	26.69[ $\sim$ NPDS]	GPDS(122)	32.44	25.96[ $\sim$ NPDS]
GPDS(256)	27.20	26.42	GPDS(126)	32.95	25.93[ $\sim$ N3SS]
4SS	16.07	26.18	GPDS(252)	34.97	25.80[ $\sim$ 4SS]
N3SS	13.67	25.98	GPDS(256)	36.17	25.72
3SS	12.89	25.43	3SS	18.62	25.66

TABLE III  
SPEEDUP RATIO AND PSNR(DB) PERFORMANCE ON DIFFERENT BMA AND GPDS WITH DIFFERENT SPEEDUP FACTOR K.

proposed GPDS has PSNR performance very close to NPDS at  $k=122$  with computational reduction of 1.82-2.37 times as compared to NPDS. Thus, GPDS is very suitable for a wide range of video applications such as speed-oriented video conferencing and high quality video coding.

## VI. ACKNOWLEDGMENTS

The work described in this paper was substantially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. 9040423].

## REFERENCES

- [1] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing", in *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, Nov 1981, pp. G5.3.1-G5.3.5.
- [2] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 438-443, Aug 1994.
- [3] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 313-317, Jun 1996.
- [4] C. K. Cheung and L. M. Po, "Normalized partial distortion search algorithm for block motion estimation", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 417-422, Apr 2000.
- [5] C. D. Bei and R. M. Gray, "An improvement of the minimum distortion encoding algorithm for vector quantization", *IEEE Trans. Commun.*, vol. COM-33, pp. 1132-1133, Oct 1985.