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 absent 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 (NPDSS) 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 NPDSS 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. 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 (3NSS) [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 (NPDSS) [4]. NPDSS 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 NPDSS by introduction of progressive partial 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) \( d_{(x,y), (u,v)}(a_{x,y}, b_{u,v}) \) with \((u,v)\) displacement from \( a_{x,y} \) is, \( q=r=1 \), using sum absolute error (SAE):

\[
d_d(x,y;u,v,q,r,M,M) = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} |a_{x+qi,y+rv} - b_{x+ui,y+vr}| \tag{1}
\]

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, NPDSS 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, \( D_{\text{MIN}} \). Experimental results in [4] show that NPDSS 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 by 16 pixel-operations for each impossible CMV. However, NPDSS limits the maximum computational reduction to \( \frac{\text{no. of pixels at the first partial distortion}}{M^2} = 16 \) 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 NPDSS, such as the...
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 $M^2 = 64$ 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 $M \times \frac{M}{H}$ to $M \times \frac{M}{H}$. In order to support $H$ equal-sized progressive partial distortions. Each partial distortion $d_{p,h}$ is firstly sampled by horizontal $s$ and vertical $l$ pixels, respectively. The first partial distortion $d_1$ is subsampled by horizontal $k$ and vertical $l$ pixels, respectively. Thus, $d_1$ is replaced by $H d_{p,h}$ for $1 \leq h \leq H$ and the two types of partial distortions are defined as in Eqn(2) and Eqn(3), respectively.

$$d_{p,h} = d_{(x+y)+h,y+4P_{i}+n,k,l,M} \bigg|_{p=1,1 \leq h \leq H} \tag{2}$$

$$d_p = \frac{d_{p,h}}{P} \bigg|_{2 \leq p \leq P} \tag{3}$$

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).

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

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 $D'_p$ and Eqn(6) for its normalization $D'_{NORM}(g)$ cumulated to $n$ pixels after PPD is introduced.

$$D'_p = \sum_{k=1}^{p} d'_k \tag{5}$$

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

$$D'_{NORM}(g) = \frac{M^2}{n} D'_p \tag{6}$$

where $n = \left\{ \begin{array}{ll}
  \frac{M^2}{m} & \text{for } p = 1, 1 \leq g \leq H \\
  p \times M^2 / P & \text{for } 2 \leq p \leq P, (H+1) \leq g \leq G
\end{array} \right.$

Fig. 2 shows different subsampling patterns on the first partial distortion. Regular subsampling $\psi (\psi \in \Psi; \Psi = \Phi \cup \Delta)$ with $i$ pixel(s) taken out at a time from a group of $S=M^2 / P$ pixels has combinations of $S = \sum_{i=1}^{S} \prod_{j=0}^{i-1} (S-jC_i)$, where $C_i = n^i / (n-i)!$. Thus, the total combination $\Phi$ including the remaining progressive regular and irregular subsampling $\Delta$ is very large. Regularity favors practical implementations.

$$D_{NORM}(H) = D_{NORM}(1) \text{ as shown in Eqn(7).}$$

$$D'_{NORM}(g) = \left\{ \begin{array}{ll}
  D_{NORM}(h) & \text{for } g = 1, 1 \leq h \leq H \\
  D_{NORM}(p) & \text{for } 2 \leq p \leq P, (H+1) \leq g \leq G
\end{array} \right. \tag{7}$$

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
such as 3SS, N3SS and 4SS.

III. GENERALIZED PARTIAL DISTORTION COMPARISON

GPDS is to provide generalization of the normalized cumulative partial distortion $D_{\text{NORM}}$ so that the NPDS matching criteria against $D_{\text{MIN}}$ 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:

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

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 $D_{\text{NORM}}(n)$ provides matching criteria for optimal results as in PDS when $k=1$, i.e. $f(n,k)|_{k=1}=M^2$. When $k<\infty$, GPDS gives the performance as in NPDS, i.e. $f(n,k)|_{k=\infty}=n$. 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 $D_{\text{NORM}}(n)$ 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 (360x240) 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 prediction quality in terms of PSNR and MSE, and computational complexity are compared.

Results using various PDD can be referred to section II, Table I and II. In general, GPDS gives 19.37 to 62.39 times speedup with 0.30-0.97dB 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
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 PPD 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 $M^2D_n\rightarrow f(n)D_{MIN}$. 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 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