ENHANCED PREDICTIVE MOTION VECTOR FIELD ADAPTIVE SEARCH TECHNIQUE (E-PMVFAST) – BASED ON FUTURE MV PREDICTION

Hoi-Ming Wong, Oscar C. Au, Chi-Wang Ho, Shu-Kei Yip

The Hong Kong University of Science and Technology, Hong Kong
Email: hoimingw@ust.hk, eeau@ust.hk, jodyho@ust.hk, sukiyip@ust.hk

ABSTRACT

Motion Estimation (ME) is a core part of most modern video coding standard, and it directly affects the compression efficiency and visual quality of a video. If Full Search (FS) algorithm is used, ME could takes over 70% of computational power. Many algorithms such as TSS and PMVFAST, have been developed to achieve great speed up for ME. In this paper, a new algorithm Enhanced-PMVFAST (E-PMVFAST) is proposed, which performs better than most if not all other existing algorithms in terms of speed up factor while keeping the similar PSNR with FS. Our experiments have also verified the robustness of the proposed E-PMVFAST algorithm.

1. INTRODUCTION

Most of state of art video compression standard, such as MPEG4 and ITU JVT/H.264 [1], use Block Matching Motion Estimation (BMME) to exploit temporal correlation between frames in a video and then achieve high compression by reducing this redundancy. The most common distortion measurement between the regarding block and referenced block in ME is Sum of Absolute Difference (SAD), for an NxN block, which is defined as:

$$SAD(mvx, mvy) = \sum_{m=0,n=0}^{N-1} \left| F_t(x+m, y+n) - F_{t-1}(x+m+mvx, y+n+mvy) \right|$$

where F_t is the current frame, F_{t-1} is the previous frame and (mvx, mvy) represents the current motion vector (MV).
For a frame with width = X, height = Y, Search Range = W and the block size is NxN, the total number of search points of ME for this frame equals to:

$$\left(\frac{X}{N}\right)\left(\frac{Y}{N}\right)(2W+1)^2 = 1673100$$

where X = 352, Y=288, N=16 and W = 32 in this case. This is a huge number that consume the largest portion of computation power in a video encoder. Many fast algorithms [2] – [9] have been proposed to reduce the number of search points in ME, such as *Three-Step Search (TSS) [11], 2D log Search [12], New Three-Step Search (NTSS) [3], MVFAST [7], and PMVFAST [2]*. *MVFAST* and *PMVFAST* significantly outperform over the first two algorithms as they performed center bias ME using median motion vector (MV) predictor as search center and hence reduced the number of bits that used to encode the MV by smoothing the motion vector field. However, both of these algorithms did not consider the change of the median MV predictor for the next block due to the selection of current MV which can affect the smoothness of the whole motion vector field. In this paper we proposed a new algorithm named *Enhanced Predictive Motion Vector Field Adaptive Search Technique (E-PMVFAST)*, different from *PMVFAST*, not only did *E-PMVFAST* consider the motion field smoothness by involving the current median MV predictor, but also involves the estimated median MV predictor of the future coding block.

In section 2, *PMVFAST* will be briefly introduced. The sub-optimal motion vector field will be discussed in section 3. We will describe the *E-PMVFAST* in section 4. Section 5 and 6 are the simulation result and conclusion.

2. PREDICTIVE MOTION VECTOR FIELD ADAPTIVE SEARCH TECHNIQUE (PMVFAST)

In [2] *PMVFAST* algorithm was introduced. This algorithm has significantly improvement on *MVFAST* and other fast algorithms, and thus was previously accepted into MPEG standard [10]. At the beginning, *PMVFAST* initial a set of MV predictors, includes median, zero, left, right, topright and previous MV predictor as shown in Figure 2. And then compute the RD Cost [13] for each of these predictors using the following cost function:

$$J(m, \lambda_{motion}) = SAD(s, c(m)) + \lambda_{motion}(R(m-p)) \tag{1}$$

where s is the original video signal and c is the referencd video signal, m is the current MV, p is the median MV predictor of the current block, λ_{motion} is the lagrange multiplier and R(x) represents the bits used to encode the motion information. The next step is to select the MV predictor that has minimum cost, then performs large or small diamond search based on the value of the minimum RD cost obtained from the MV predictors.

0-7803-9332-5/05/$20.00 ©2005 IEEE

3. THE SUB-OPTIMAL MOTION VECTOR FIELD

For each frame in a video, there must exists a motion vector field that globally minimizes the total RD Cost for the whole frame, that is, to minimize:

$$OptimalCost = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left[SAD(s_{i,j}, c(m_{i,j})) + \lambda_{i,j}(R(m_{i,j} - p_{i,j}))\right]$$

(2)

Where (i, j) represents a block in a frame with MxN blocks. For fixed QP, $\lambda_{i,j} = \lambda$ = CONSTANT, and

$$p_{i,j} = median(m_{i-1,j}, m_{i,j-1}, m_{i+1,j-1})$$

(3)

However, obtaining the optimal cost for whole frame has exponential order of computational complexity, which is not practical to do so. So the scope of E-PMVFAST is restricted to optimizing the MVs of consecutive blocks instead of the whole frame. The formula (2) and (3) show that the choice of current MV directly affects the RD cost of the next block. For fixed RD cost of current block, minimizing:

$$R(|m_{i+1,j} - p^{+}_{i,j}|) \quad (4)$$

will minimize the RD cost for next block where $p^{+}_{i,j}$ is the estimated median MV predictor for next block, i.e.

$$p^{+}_{i,j} = median(m_{i,j}, m_{i+1,j-1}, m_{i+2,j-1}) \quad (5)$$

Since R(x) is an injective monotonic increasing function *[13]* for x > 0, this leads to the minimization is equivalent to minimizing:

$$E[|m_{i+1,j} - p^{+}_{i,j}|] \quad (6)$$

$m_{i+1,j}$ is unknown, but the above formula can be approximated by $E[|m_{i,j} - p^{+}_{i,j}|]$, figure 1 shows the distribution of $|m_{i+1,j} - p^{+}_{i,j}|$ (shown in solid line) and $|m_{i,j} - p^{+}_{i,j}|$ (shown in dotted line) for Foreman sequences, we can conclude that they are roughly in the same distribution with different variance. As a result, $R(|m_{i+1,j} - p^{+}_{i,j}|)$ can be approximated by W* $R(|m_{i,j} - p^{+}_{i,j}|)$, where W > 0. Let s be $|m_{i,j} - p^{+}_{i,j}|$, we can model s to be a Single side Laplacian random variable for s >= 0, i.e.

$$P(s) = \alpha e^{-\alpha s}$$

(7)

Where α is the Laplacian parameter, and σ^2 is the variance, E[s] equals to:

$$E[s] = \int_0^{\infty} sP(s)ds = \frac{1}{\alpha}$$

(8)

and,

$$\sigma^2 = \frac{1}{\alpha}\left(\frac{1}{\alpha} - 1\right) \quad (9)$$

The above equations show that minimizing equation (6) is equivalent to minimizing the variance of s for the next block while selecting the current MV. When $m_{i,j} \to m_{i+1,j-1}$ and $m_{i,j} \to m_{i+2,j-1}$, the variance of $p^{+}_{i,j}$ is minimized, and hence the expected RD Cost for the next block is minimized.

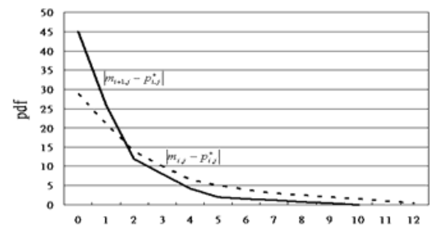

Fig. 1– the distribution of $|m_{i+1,j} - p^{+}_{i,j}|$ *and* $|m_{i,j} - p^{+}_{i,j}|$

4. ENHANCED PREDICTIVE MOTION VECTOR FIELD ADAPTIVE SEARCH TECHNIQUE (E-PMVFAST)

E-PMVFAST does not use the left, right, topright and zero MV predictors that are used in PMVFAST, because these predictors may belong to some outlier objects that far away from the median MV predictor, and hence causes the current MV going far away from the median MV predictors that cause large MV cost. Section 3 showed the current MV selection criteria of *E-PMVFAST. E-PMVFAST* introduced the Future median MV predictor as a new search center and a new weighed RD Cost function that involves $p^{+}_{i,j}$. The remaining part of *E-PMVFAST* is basically similar to *PMVFAST*, which involves Small or Large Diamond Search.

Step 1: Obtain the median MV predictor $p_{i,j}$ (MedianMV), previous MV (PreMV).

Step 2: Compute the Future median MV predictor $p^{+}_{i,j}$ (FMedianMV), which is equal to Median(MedianMV, TopRighMVt, TopRightRightMV).

Step 3: When the current frame is the first P frame, then PreMV is invalid. When current block is at top or right boundary of the frame, then FMedianMv is invalid.

Step 4: Compute Costs using all of the valid MV predictors, if |MVx - MedianMVx | <= 4 integer pixel and |MVy - MedianMVy | <= 4 integer pixel, then cost = SAD + R(MV – MedianMV); otherwise, cost = SAD + w1 * R(MV – MedianMV) + w2 * R(MV – FMedianMV).

Step 5: Select the MV with smallest cost, perform 1 step of Small Diamond Search.

Step 6: If cost < T1, go to step 9.

Step 7: If cost < T2, perform Small Diamond Search, stop, go to step 9.
Step 8: Perform Large Diamond Search, go to step 9.
Step 9: Select the MV with smallest cost.

where T1 is the minimum SAD value of left, top, and topright blocks. And T2 equals to T1 + 256.

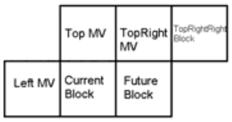

Fig. 2– Left, Top, TopRight, TopRightRight MV, current block and future block.

5. SIMULATION RESULTS

The proposed E-PMVFAST was embedded into H.264 reference software JM9.2 [13], and tested using various QP, video sequence, different resolution (CIF and QCIF), and different search range (search range = 32 for CIF and 16 for QCIF). Table 1 and 2 show the simulation results, PSNR diff. and Bitrate diff. are the change of PSNR and bitrate comparing with full search. The simulation result shows in QCIF, the *E-PMVFAST* has a similar bitrate and PSNR comparing with full search and *PMVFAST* on average and much better than *TSS*. The *E-PMVFAST* also has larger speed up than *PMVFAST* and *TSS*. The simulation also shows the robustness of *E-PMVFAST*, for different video sequences and bitrate, E-PMVFAST has very stable performances. The most important feature of *E-PMVFAST* is that its motion vector field is very smooth, so that the motion vectors can represent the objects' movement more accurate than other fast motion estimation algorithms. Figure 3a and 3b shows the motion vector field of *E-PMVFAST* is significantly smoother than that in *PMVFAST*, especially in the circled regions. This information is very useful for classifying the motion content of a video for use of perceptual transcoding, rate control, multiple block size motion estimation, multiple reference frame motion estimation, and so on.

6. CONCLUSION

This paper proposed a new fast motion estimation algorithm named Enhanced Predictive Motion Vector Field Adaptive Search Technique (*E-PMVFAST*), which achieved the highest speed motion estimation with very good quality comparing with *PMVFAST* and other popular fast motion estimation algorithms.

7. ACKNOWLEDGEMENT

This work has been supported in part by the Research Grants Council of the Hong Kong Special Administrative Region, China (project no. HKUST6203/02E), the Innovative Technology Commission (project no. ITS/122/03) and HKUST (project no. DAG03/04.EG10).

7. REFERENCES

[1] Joint Video Team of ITU-T and ISO/IEC JTC 1, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)," *document JVT-G050r1*, May 2003.

[2] A. M. Tourapis, O. C. Au, and M. L. Liou, "Predictive Motion Vector Field Adaptive Search Technique (PMVFAST)," to be presented in *ISO/IEC JTC1/SC29/WG11 MPEG2000, Noordwijkerhout*, NL, March'2000.

[3] R. Li, B. Zeng, and M.L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. On Circuits and Systems for Video Technology,* vol. 4, no. 4, pp. 438-42, Aug'94.

[4] Z.L. He and M.L. Liou, "A high performance fast search algorithm for block matching motion estimation," *IEEE Trans. on Circuits and Systems for Video Technology,* vol.7, no.5, pp.826-8, Oct'97.

[5] A.M. Tourapis, O.C. Au, and M.L. Liou, "Fast Motion Estimation using Circular Zonal Search", *Proc. of SPIE Sym. Of Visual Comm. & Image Processin*, vol.2, pp.1496-1504, Jan. 25-27, '99.

[6] A. M. Tourapis, O. C. Au, M.L. Liou, G. Shen, and I. Ahmad, "Optimizing the Mpeg-4 Encoder – Advanced Diamond Zonal Search", in *Proc. of 2000 IEEE Inter. Sym. on Circuits and Systems*, Geneva, Switzerland, May, 2000.

[7] K.K. Ma and P.I. Hosur, "Performance Report of Motion Vector Field Adaptive Search Technique (MVFAST)," in *ISO/IEC JTC1/SC29/WG11 MPEG99/m5851*, Noordwijkerhout, NL, Mar'00.

[8] A.M. Tourapis, O.C. Au, and M.L. Liou, "Fast Block-Matching Motion Estimation using Predictive Motion Vector Field Adaptive Search Technique (PMVFAST)," in *ISO/IEC/ JTC1/SC29 /WG11 MPEG2000/M5866, Noordwijkerhout, NL*, Mar'00.

[9] Implementation Study Group, "Experimental conditions for evaluating encoder motion estimation algorithms," in ISO/IEC JTC1/SC29/WG11 MPEG99/n3141, Hawaii, USA, Dec'99 (updated and modified January 2000).

[10] "MPEG-4 Optimization Model Version 1.0", in *ISO/IEC JTC1/SC29/WG11 MPEG2000/N3324*, Noordwijkerhout, NL, Mar'00.

[11] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," *Proc. Nat. Telecommun. Conf., New Orleans*, LA, pp. G5.3.1-G5.3.5, Dec'81.

[12] J.R. Jain and A.K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. On Communications*, vol. COM-29, pp.1799-808, Dec'81.

[13] JVT reference software JM9.2 for JVT/H.264 FRext.

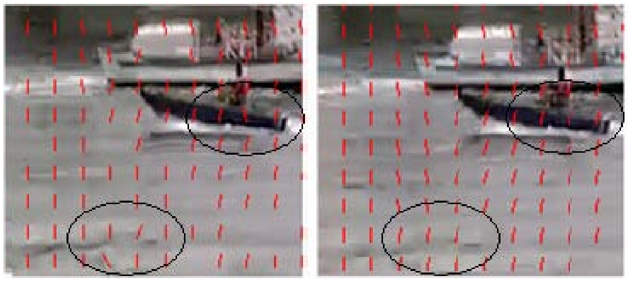

Fig. 3a – Motion vector field smoothness comparison between PMVFAST (Left) and E-PMVFAST (Right).

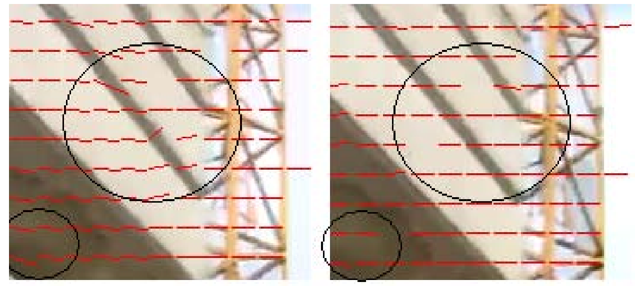

Fig. 3b – Motion vector field smoothness comparison between PMVFAST (Left) and E-PMVFAST (Right).

Table 1 –Comparison between FS, TSS, PMVFAST, and E-PMVFAST for CIF sequences

	FULL Search			TSS			PMVFAST			E-PMVFAST		
Sequences	QP	PSNR	Bitrate	PSNR diff	Bitrate diff	Speed up	PSNR dif	Bitrate diff	Speed up	PSNR diff	Bitrate diff	**Speed up**
Foreman	28	35.99	4866496	-0.07	29.89%	176.04	-0.02	1.56%	248.98	-0.02	0.86%	**275.53**
	30	34.64	3436840	-0.08	31.26%	176.04	-0.03	1.35%	241.31	-0.03	0.86%	**268.51**
	32	33.36	2471608	-0.1	30.78%	176.04	-0.05	1.37%	234.74	-0.05	0.82%	**262.23**
	34	32.23	1842904	-0.09	28.04%	176.04	-0.03	1.22%	230.14	-0.07	1.08%	**258.28**
	36	31.06	1366224	-0.13	24.68%	176.04	-0.06	1.36%	226.38	-0.05	1.06%	**255.50**
	38	29.92	1053768	-0.12	21.12%	176.04	-0.09	1.45%	223.10	-0.1	0.80%	**253.59**
	40	28.89	847008	-0.13	17.99%	176.04	-0.09	1.77%	221.14	-0.12	0.78%	**252.26**
akiyo	28	39.5	819840	-0.15	28.76%	176.04	-0.02	0.02%	409.14	-0.02	-0.07%	**449.32**
	30	38.17	585312	-0.15	27.51%	176.04	-0.03	0.11%	379.83	-0.01	0.34%	**422.17**
	32	36.84	423392	-0.11	26.92%	176.04	-0.04	-0.03%	346.52	-0.01	-0.54%	**396.21**
	34	35.68	315600	-0.14	23.78%	176.04	-0.01	0.13%	323.51	0	0.61%	**378.72**
	36	34.45	231824	-0.17	20.03%	176.04	-0.03	1.05%	299.70	-0.04	0.39%	**355.36**
	38	33.12	176248	-0.21	18.01%	176.04	-0.01	0.81%	279.13	-0.04	0.39%	**332.95**
	40	32.01	141056	-0.18	11.26%	176.04	-0.05	0.29%	265.58	-0.05	0.42%	**318.55**
coastguard	28	34.34	10154944	-0.04	8.16%	176.04	0	-0.02%	233.07	0	-0.11%	**269.68**
	30	32.8	7126176	-0.04	9.49%	176.04	0	-0.05%	230.71	0	0.11%	**267.69**
	32	31.34	4904088	-0.04	10.51%	176.04	0.01	-0.05%	228.94	0	-0.18%	**266.10**
	34	30.08	3380472	-0.04	12.12%	176.04	0	0.10%	227.64	-0.01	0.13%	**265.05**
	36	28.74	2194280	-1.04	13.15%	176.04	-0.01	0.00%	226.02	0	0.04%	**263.84**
	38	27.55	1452880	-0.04	14.23%	176.04	-0.03	0.31%	224.81	-0.03	0.19%	**262.99**
	40	26.55	1009152	-0.03	14.60%	176.04	-0.04	0.08%	224.38	-0.04	-0.04%	**263.84**
News	28	37.78	2298128	-0.14	12.49%	176.04	-0.05	0.51%	346.04	-0.03	0.15%	**377.23**
	30	36.36	1748048	-0.16	12.88%	176.04	-0.02	0.53%	322.06	-0.01	0.58%	**357.56**
	32	34.98	1328088	-0.15	12.59%	176.04	-0.02	0.40%	303.69	-0.03	0.48%	**341.92**
	34	33.64	1028192	-0.08	11.25%	176.04	-0.04	0.43%	284.55	-0.01	0.33%	**327.79**
	36	32.25	761744	-0.14	10.60%	176.04	-0.02	0.92%	271.80	-0.05	0.87%	**312.10**
	38	30.85	578960	-0.13	9.78%	176.04	-0.06	0.89%	262.68	-0.04	0.86%	**302.93**
	40	29.65	448472	-0.06	9.94%	176.04	-0.01	0.87%	252.46	-0.02	0.98%	**296.89**
Average				**-0.14**	**17.92%**	**176.04**	**-0.03**	**0.62%**	**270.29**	**-0.03**	**0.44%**	**309.10**

Table 2 –Comparison between FS, TSS, PMVFAST, and E-PMVFAST for QCIF sequences

	FULL Search			TSS			PMVFAST			E-PMVFAST		
Sequences	QP	PSNR	Bitrate	PSNR diff	Bitrate diff	Speed up	PSNR dif	Bitrate diff	Speed up	PSNR diff	Bitrate diff	**Speed up**
Hall	28	37.09	606800	-0.01	3.13%	45.39	-0.05	-0.33%	62.34	-0.01	-0.18%	**81.93**
	30	35.57	460648	-0.03	3.48%	45.39	-0.03	0.17%	60.80	-0.01	-0.05%	**80.81**
	32	33.98	360080	0.01	3.53%	45.39	0.01	-0.30%	59.98	0.02	-0.06%	**79.94**
	34	32.55	276912	-0.08	3.68%	45.39	-0.05	-0.45%	59.12	-0.05	-0.56%	**79.25**
	36	31	207304	-0.02	4.21%	45.39	0	-0.29%	58.59	0.01	-0.40%	**79.01**
	38	29.62	158448	0	2.92%	45.39	-0.02	-0.93%	58.04	-0.04	-0.78%	**78.68**
	40	28.28	123784	-0.06	2.07%	45.39	-0.01	-0.31%	57.60	-0.03	-0.24%	**78.37**
mobile	28	32.78	4414304	-0.1	40.07%	45.39	0	-0.10%	66.43	0	-0.07%	**87.28**
	30	30.88	3153648	-0.08	46.01%	45.39	0	-0.09%	63.49	0	0.07%	**85.25**
	32	29.17	2224192	-0.06	51.74%	45.39	-0.01	-0.30%	62.24	-0.01	-0.38%	**84.41**
	34	27.63	1587296	-0.03	52.88%	45.39	0	-0.10%	61.10	0	0.03%	**83.54**
	36	26.03	1068160	-0.08	56.64%	45.39	-0.01	0.27%	60.69	0	0.13%	**83.24**
	38	24.62	759056	-0.11	48.68%	45.39	0	-0.02%	60.47	0.01	-0.51%	**82.80**
	40	23.39	554968	-0.15	39.40%	45.39	-0.01	-0.18%	60.30	-0.02	0.06%	**82.79**
coastguard	28	33.92	2144008	-0.05	26.05%	45.39	-0.01	0.15%	87.44	0	0.05%	**106.36**
	30	32.32	1464496	-0.04	28.93%	45.39	0.01	-0.09%	77.95	0	-0.19%	**100.55**
	32	30.87	997800	-0.04	31.68%	45.39	-0.01	0.15%	71.93	0	-0.35%	**95.40**
	34	29.63	696728	-0.07	33.81%	45.39	-0.02	-0.56%	69.11	-0.01	-0.37%	**90.47**
	36	28.31	465640	-0.07	32.56%	45.39	-0.03	0.35%	66.91	-0.03	0.10%	**89.29**
	38	27.09	326424	-0.06	29.07%	45.39	-0.01	-1.32%	64.17	-0.01	-0.96%	**86.97**
	40	26.03	234392	-0.04	26.11%	45.39	-0.01	0.29%	62.86	-0.03	0.30%	**85.44**
Average				**-0.06**	**26.98%**	**45.39**	**-0.01**	**-0.19%**	**64.36**	**-0.01**	**-0.21%**	**85.80**