14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

REDUCED COMPUTATION USING ADAPTIVE SEARCH WINDOW SIZE FOR H.264 MULTI-FRAME MOTION ESTIMATION

Liang-Ming Ji and Wan-Chi. Siu

Department of Electronic and Information Engineering,
Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
Facsimile: +852 2362 8439
Email: jiliangming@msn.com enwcsiu@polyu.edu.hk

Abstract

Motion estimation of the H.264 takes the advantage of multi-reference frames to select the best matching position. The full process of searching up to 5 past reference frames leads to high computational complexity and casts a heavy burden on the encoder. After a careful analysis, it is found that the motion vector obtained in the 1st reference frame can be used as a guide to adaptively set an appropriate size of the search window for the rest of the reference frames. Meanwhile, if several small blocks have the same motion vector, the combined large block can utilize such a motion vector as an initial point for further refinement within a small search window size. Results of our experimental work using the JM9.6 show that the proposed algorithm can reduce the computational complexity to 10% of the original full search and meanwhile keep the PSNR drop less than 0.02dB with less than 1% increase in bitrate.

I. INTRODUCTION

By making use of seven variable block sizes, the newly adopted H.264/AVC [1] video standard divides a frame flexibly into blocks with different sizes for efficiently video coding. The H.264/AVC is the state-of-the-art video-coding standard developed by the Joint Video Team (JVT), which gathered experts from ISO/IEC MPEG-4 Advanced Video Coding (AVC) and ITU-T H.264. The sub-pixel accuracy, including half-pixel and quarter-pixel, refines the motion estimation. 4x4 Integer transform is used instead of the traditional 8x8 DCT. All these new features significantly improve the H.264 in bit-rate reduction. Compared to the MPEG-4 advanced simple profile, up to 50% of bit-rate reduction can be achieved. Meanwhile, rather than using only one previous frame as the reference, the H.264 allows the encoder to use up to five previous frames for motion estimation. Searching for the most matching block in the previous several frames by the full searching approach inevitably increases the complexity linearly, in that for each reference frame, the motion estimation is performed and the cost is calculated to make the decision on whether this is the most matching block.

Obviously, making full search to all candidate frames is able to achieve the best coding result, but the operational complexity is proportional to the number of searched frames, which results in a high computational cost. Besides, the values of prediction errors depend on the nature of the video sequences. Sometimes much effort has to be made to obtain an insignificant gain. In this case, much computation are wasted. Some work has been done to reduce the complexity caused by the multiple reference frames. These methods can be classified into 2 categories:

A) Selective searching point

This approach makes a sub-sample of blocks within the search range in order to achieve a faster search convergence. The main difference between this approach and those conventional block-matching algorithms such as diamond search or cross-diamond search is that its scope covers the information obtained from several reference frames instead of just looking at the local statistic, so it can be viewed as an extension of the fast single frame algorithm. For example, the simplex minimization search was extended to multiple reference motion estimation in [2]. Ting, Lam and Po [3] adopted a novel recent-biased spiral-cross search pattern to sub-sample the 3-dimensional memory space as a whole. Chan and Siu [4] suggested that feature points within a video frame are allowed to locate the initial search point for motion vector search. Also, adaptive partial distortion search could be used to further eliminate the impossible points as soon as possible [5].

B) Selective searching frame

This approach sub-samples frames from the memory buffer to eliminate unrelated reference. This category can be further divided into 2 sub-divisions.

i) Early termination of the full search process using some criteria: References [6] and [7] proposed to set the cost as a threshold to make decision on whether to do motion estimation on reference frame 4 and 5. These methods exploit the decreasing temporal correlation between consecutive frames; once they find that the remaining frames may not be contributive, they stop the motion estimation.

ii) Selecting candidate frames from 5 reference frames: One of the easiest approaches is to use the same reference frame as its neighboring macroblocks' reference frames to encode the current macroblock [8]. In this case some reference frames may be skipped. This method is simple in implementation, however the result is not totally satisfactory. Applying a center-biased frame selection path to identify the ultimate reference frame from all candidates was proposed in [9]. This approach gave focus on the distribution of motion vectors and used the cross pattern to find the frame number containing the best point among all 5 references and then performed full search on the selected frame.

II. ANALYSIS OF MULTIPLE REFERENCE USAGE

In multiple reference frame motion estimation, motion estimation is performed for each reference frame. In Fig.1, Ref1 is the closet reference frame to the current frame, while Ref5 is the farthest. The verification model selects the best motion vector based on the criterion of minimized cost, which can be defined as

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

where $m = (m_x, m_y)$ is the motion vector, $p = (p_x, p_y)$ is the prediction for the motion vector and λ_{motion} is the Lagrange multiplier. The term $R(m-p)$ represents the bits to be used to encode the motion information and $R(REF)$ gives the bits indicating which reference picture will be used.

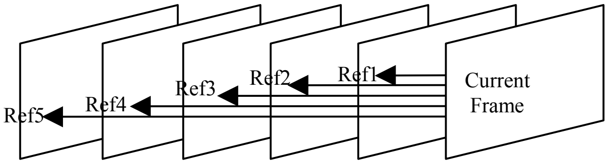


Fig. 1. Multiple reference frame search

Table I shows the bits allocation for motion information, (m-p). Note that it is in the unit of quarter-pixel, which means that one pixel unit difference is regarded as 4. It is straightforward to handle the bit allocation, *R(REF),* for indicating the reference frame to be used. The closet reference frame needs 1 bit and the next 2 reference frames need 3 bits, and the farthest 2 frames need 5 bits as shown in table II. By comparison, it is found that the overhead for coding the motion vector is more expensive than that for indicating the reference frame, since even one pixel displacement of the predicted motion vector will result in an overhead of 7 bits, while the farthest reference only takes 5 bits. For H.264, the cost is calculated by multiplying the bits with λ, which is defined as $\sqrt{0.85 * 2^{(QP-12)/3}}$.

Index	Difference between best MV and predicted MV (m-p)	Bits
0	0	1
1	±1	3
2	±2, ±3	5
3	±4, ±5, ±6, ±7	7
4	±8, ±9, ±10, ±11, ±12, ±13, ±14, ±15	9
…	…	…
N	$\pm 2^{N-1} \sim \pm 2^{N} - 1$	2N + 1

Table I Bits allocation for (m-p)

Index	Reference frame No.	Bits
0	0	1
1	1, 2	3
2	3, 4	5

Table II Bits allocation for reference frames

In order to understand better the multi-frame motion estimation, preliminary experiments were done. Experimental results in Table III were generated by JM9.6. We encoded 50 frames for 16 video sequences in CIF (352x288) format. The encoding scheme is IPPPP…. These testing sequences cover large, medium and small motion activities. Obviously, the first reference frame is most likely to be chosen as the reference. It is reasonable because the temporal correlation between references decreases when the interval increases. It is concluded from the table that about 85% of the final choice of motion vector resides in the 1st reference frame, which is the closet frame.

Moreover, sequences with large motion activities such as "mobile" have a stronger dependency on frames other than the 1st reference, while sequences with small motion activities almost exclusively use the 1st frame as the reference. Based on such motion vector magnitude information, we can actually decide whether it is necessary to use more reference frames. Table IV shows the results of coding videos using 1 to 4 reference frames as compared with using 5 references, from which we can understand how well the number of frames used for encoding affects the bitrate, PSNR and complexity of realization. Note that the complexity refers to the numbers of the addition operations for calculating the SAD, which accounts for the high cost in motion estimation. It is found that the complexity increases linearly with the number of reference frames used. The more reference frames are used for encoding, the better is the PSNR and the smaller is the bitrate.

Video Sequences	Ref1 %	Ref2 %	Ref3 %	Re4 %	Ref5 %
mobile	38.51	16.1	19.2	14.2	12
tempete	54.92	11	14.5	10.4	9.19
akiyo	97.03	1.14	0.72	0.55	0.56
hall	95.64	1.36	1.22	0.84	0.94
M & D	94.22	1.87	1.89	0.95	1.08
silent	95.04	1.81	1.1	1.02	1.03
salesman	93.59	4.57	0.71	0.68	0.46
paris	92.05	3.71	2.18	1.15	0.91
news	97.29	1.2	0.74	0.43	0.34
highway	87.51	4.27	3.31	2.58	2.34
foreman	78.85	8.07	5.66	3.64	3.79
coastguard	91.84	3.61	1.88	1.71	0.95
bridge far	99.97	0	0.01	0.01	0.02
bridge close	99.91	0.03	0.06	0	0
bus	57.07	19.4	13.1	6.81	3.63
carphone	76.83	7.88	6.53	4.53	4.22
AVERAGE	***84.39***	***5.37***	***4.55***	***3.1***	***2.59***

Table III Reference usage %

Reference frames used	% increase of Bitrate	Drop in PSNR dB	Complexity %
4	0.99	0.011	79
3	2.43	0.027	58
2	5.9	0.051	38
1	11.9	0.079	18

Table IV Results by using different reference frames

Video Sequences	SW = 8		SW = 32	
	% increase of Bitrate	Drop in PSNR dB	% increase of Bitrate	Drop in PSNR dB
mobile	0.20	0.006	0.02	-0.007
tempete	0.20	0.013	-0.36	-0.01
akiyo	-0.11	0.002	0.13	0.001
hall	0.47	0.023	-0.59	-0.018
M & D	0.36	0.005	0.16	0.005
silent	1.80	0.025	-0.76	-0.011
salesman	-0.06	-0.005	-0.46	-0.017
paris	0.14	0.003	-0.06	-0.005
news	-0.20	0.046	0.36	-0.005
highway	1.59	0.033	0.03	0.005
foreman	0.54	0.019	0.25	-0.021
coastguard	-0.11	0.002	-0.32	0.008
bridge far	-0.32	0.004	-0.84	0.001
bridge close	0.00	0	-0.01	0
bus	4.62	0.053	-1.22	-0.024
carphone	0.88	0.032	-0.66	-0.017
AVERAGE	***0.63***	***0.016***	***-0.27***	***-0.007***

Table V Results by setting SW = 8 or 32 compared with Results by setting SW = 16

Further simulation results in Table V show that when reducing the size of search window (SW), there is little degradation for some of the sequences. For example, "bridge_far" and "bridge_close" have insignificant PSNR drop. A large window size always ensures a good motion estimation, however for some sequences with small motion activities, smaller search window size does not necessarily mean a poor motion estimation.

III. ADAPTIVE SEARCH WINDOW SIZE

The motion information of an object can be considered as a constant velocity movement in one direction if the time interval is small. Fig.2 shows a scenario, for which an object moves in the downward direction. The motion vectors will become larger and larger when it searches reference frames farther away from the current frame. As discussed before, larger motion vectors and farther reference frames require more bits and therefore for the sake of saving bit-rate, the cost function will be in favor of choosing the smallest motion vector obtained from Ref1, which is the closet to the current frame. For this scenario, it is found that searching the rest 4 frames other than the 1st reference frame is redundant. It wastes time and requires high computational complexity while no contribution would be made. In such case, we should terminate the searching process early by eliminating the requirement to search for the rest reference frames if their motion vectors are assumed to be larger than that of Ref1.

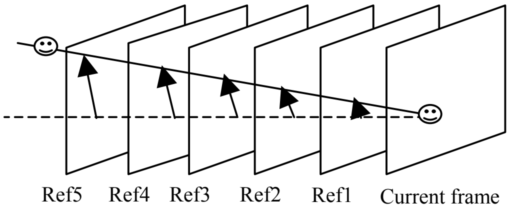


Fig. 2. Object match in multiple reference frames

Searching more than one reference frame could mean aiming at identifying the object in the several previous frames, which may yield a smaller cost than that in the 1st reference frame. Here we use the term "object" in the sense that the searching process is done successfully to find its counterpart in previous frames within a certain temporal interval. Note that the block size in H.264 can be as small as 4x4, which can finely represent a relatively small object in successive frames. Fig.3 shows the motion estimation process of matching an object in the 1st, 2nd and 3rd reference frames. MV1, MV2 and MV3 are the motion vectors for the object in different reference frames respectively. Note that among all three motion vectors, MV3 is the largest and MV2 is the smallest. This may be due to the shaking motion of the camera or the highly repetitive movement of the object.

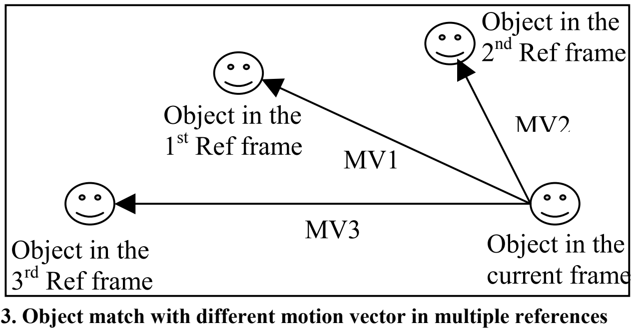


Fig.3. Object match with different motion vector in multiple references

Let us assume that the block containing the object in different frames are almost the same, so the motion estimation process will result in almost the same SAD cost. From equation (1), it is clear that in order to minimize the total cost J, we have to choose the smallest MV among all MVs. After searching through the 1st reference frame, MV1 is obtained. It is expected that MV2, MV3… should be smaller than MV1 if the search using more reference frames is meaningful. Otherwise if MV1 is finally chosen as the best motion vector, the effort of searching more frames is wasted. Initiated by the observation, we suggest using MV1 as a threshold to define a search window size for the remaining reference frames. In this example, once we obtain MV1 from the 1st reference frame, we make use of its length as the window size for searching the rest of the reference frames. With such a search window size decided, the object in the 3rd reference frame, which locates beyond the search window, can not be found. However it will not affect the final searching result since the object in the 3rd reference frame would not be chosen for MV3 is large.

Table VI lists the probability that a motion vector finally chosen as the best motion vector in reference frames 2 – 5 is larger than the candidate motion vector previously found in the 1st reference frame. From the statistics, it is found that videos with large motion tend to rely on more reference frames, while the quasi-stationary sequences seldom select larger motion vectors in reference frames 2 – 4. Besides, those larger motion vectors in a farther reference frame (Ref4 or Ref5) have a small chance to be selected than those residing in a closer reference frame (Ref2 or Ref3).

Video sequence	SW = 8				SW = 16				SW = 32			
	Ref 2	Ref 3	Ref 4	Ref 5	Ref 2	Ref 3	Ref 4	Ref 5	Ref 2	Ref 3	Ref 4	Ref 5
Mobile	16.42	23.92	17.59	14.91	16.22	23.42	17.69	14.92	15.51	22.90	17.29	14.35
Tempete	15.14	22.98	16.88	14.96	15.09	22.87	16.97	14.75	14.64	23.19	16.78	15.22
Akiyo	9.27	8.12	5.55	5.67	9.08	8.08	5.77	5.58	9.38	8.37	5.77	5.22
Hall	5.70	3.77	1.57	1.94	5.40	3.19	1.96	2.00	6.32	2.79	2.29	2.47
M & D	7.84	6.76	3.79	3.87	7.40	7.86	3.91	4.21	7.67	6.68	3.53	4.01
Silent	5.88	3.19	2.59	3.12	6.35	3.12	3.00	2.84	6.17	3.70	2.93	3.11
salesman	6.18	3.74	2.36	2.45	6.42	3.82	2.17	2.35	6.09	3.72	1.93	2.18
Paris	12.45	8.25	4.36	2.90	12.31	8.07	4.43	3.17	12.11	8.28	4.35	3.13
News	10.12	7.43	4.26	2.87	10.26	7.20	4.12	2.94	9.79	6.86	3.71	2.88
Highway	5.79	4.77	3.25	2.98	5.83	4.85	3.45	3.29	5.83	4.22	3.52	2.96
Foreman	8.05	6.45	4.38	4.18	7.71	6.47	3.98	4.15	8.22	6.47	4.38	4.33
Coast	12.56	6.48	5.57	2.69	12.34	6.40	5.76	3.07	12.20	6.43	5.61	2.97
B_ far	0.00	0.00	1.98	1.80	0.00	0.78	1.49	3.45	0.00	3.09	1.85	1.68
B_ close	25.11	43.39	0.59	0.69	24.35	43.39	0.56	0.66	23.83	40.38	0.00	1.55
Bus	25.93	8.50	2.70	1.29	19.76	13.41	7.17	3.66	18.79	12.39	7.43	5.28
carphone	6.08	4.34	3.24	2.63	6.33	4.17	2.77	2.91	5.66	3.75	3.03	2.64
Average	***10.7***	***10.1***	***5.04***	***4.31***	***10.3***	***10.44***	***5.33***	***4.62***	***10.1***	***10.20***	***5.28***	***4.62***

Table VI Probability % of MV in reference frame 2-5 larger than MV in the 1st reference frame

Note that for windows with various sizes, there is little variation in average percentage. It is due to the reason that if within a search window size in the 1st reference frame, no good match is found, the object could move very fast, out of the size of the search window. Then, most likely, in the 2nd reference frame, it is still outside the search window. The conclusion can be generalized as, if the object is within the search window in the closer reference frame, it is possible to find its equivalent initial point in a farther reference frame. On the other hand, if it cannot be found in the 1st reference frame, there is little chance that it would be found in the rest of the frames. Hence the fact that one object or an initial point either appears in all reference frames, or appears in no reference frame, makes the result almost irrelevant using the different search window sizes.

IV. PROPOSED ALGORITHM & EXPERIMENTAL RESULTS

Step 1: In this step, we consider the length of motion vector obtained in the 1st reference frame as the maximum size of the search window for the motion estimation in the rest four reference frames. All seven modes are included in the evaluation.

"Step 1" of Table VII presents the simulation results, indicating that step 1 is able to reduce the complexity into 25%, with insignificant bitrate increase and PSNR drop as compared with the previous simulation result obtained by setting the SW to 16. Step 1 reduces the searching window size for all the reference frames except that the 1st reference frame is a full search with SW equal to 16. If only one reference frame is used for encoding, the previous statistics in Table IV show that it takes 18% of the full complexity. Consequently, in step 1, 82% (100% - 18%) of the searching complexity from the rest 4 frames is now reduced to 7% (25% – 18%). For reducing the complexity further, we should do further work on the 1st reference frame.

Video	Step 1			Step 2			Proposed algorithm		
	B	P	C	B	P	C	B	P	C
mobile	0.89	0.005	28	0.13	-0.001	59	0.68	0.013	18
tempete	-0.08	0.010	28	-0.20	0.001	58	0.01	0.007	17
akiyo	0.33	0.000	22	0.23	0.008	27	0.36	0.005	7
hall	0.37	0.010	24	0.35	-0.002	20	0.42	0.007	7
M & D	0.63	-0.006	21	0.06	0.002	39	0.38	0.008	9
silent	0.21	-0.002	27	-0.02	-0.005	38	0.64	0.003	12
salesman	-0.13	-0.008	23	-0.33	0.001	31	-0.05	-0.009	8
paris	0.39	0.004	23	0.08	0.002	46	0.35	0.019	11
news	0.33	-0.001	22	-0.05	0.015	30	0.03	0.006	8
highway	1.77	0.020	27	1.54	0.031	43	2.31	0.046	12
foreman	1.23	0.017	25	0.45	0.016	61	1.21	0.011	15
coast	0.71	0.018	18	0.00	0.007	73	0.56	0.024	9
b_far	-0.04	0.000	22	-0.01	0.000	15	-0.57	0.001	4
b_ close	-0.02	0.000	21	0.00	0.000	27	-0.02	0.000	4
bus	5.72	0.046	33	-0.02	0.004	73	5.93	0.048	22
carphone	0.60	0.021	27	0.47	0.049	51	0.93	0.062	14
Average	***0.81***	***0.008***	***25***	***0.17***	***0.008***	***43***	***0.82***	***0.016***	***11***

B--- Bitrate Increase % P---PSNR Decrease dB C--- Complexity %

Table VII Experimental results of the proposed algorithm

The H.264 adopts 7 modes with different block sizes, and by exhaustive motion search, the best mode is chosen based on the criterion of the minimum cost. Usually, big block sizes are for homogenous or stationary regions, while smaller block sizes are for complex textures or objects with large motion activities [10]. If in one partition, several small blocks have the same motion vector, there is a large probability that the big block covering these small block areas will be finally encoded with the same motion vector and meanwhile the big block size will be chosen as the best mode.

Step 2 If two horizontal 8x8 blocks have the same motion vector, it is then used as a MV Predictor to encode the corresponding 16x8 block, and with a search window size equal to 2 for refinement. Similarly, the same idea is applied to form the motion vector for an 8x16 block from two vertical 8x8 blocks and a 16x16 block from four 8x8 blocks. The same idea is also used in smaller block sizes, for combining 4x4 blocks to form 4x8 or 8x4 blocks. Note that all 7 modes and 5 reference frames have been dealt with in this step. From the results in "Step 2" of Table VII, it is known that more than half of the complexity is reduced due to the smaller search window size, while the PSNR and bitrate are very close to the original levels.

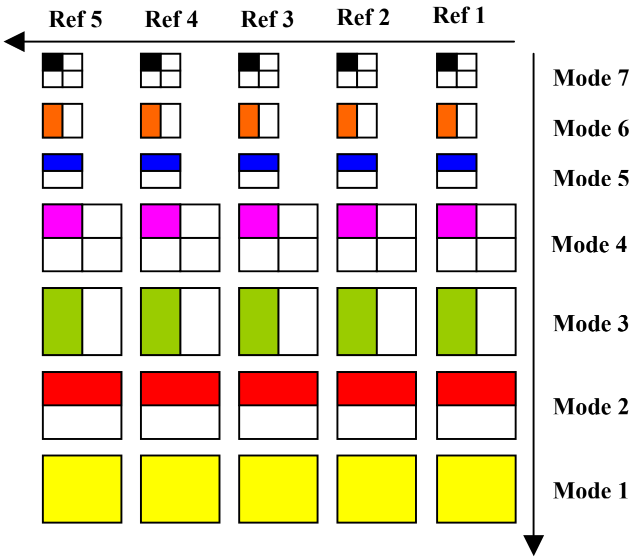


Fig.4. Motion estimation process for proposed algorithm

Proposed algorithm: The final algorithm takes the advantage of steps 1 and 2. Fig. 4 illustrates the process of the proposed algorithm, in which motion estimation is performed on small block size prior to big block size, aiming at reusing the motion vector information obtained from smaller block as shown in step 2. The method in Step 1 is also applied to determine the search window size for frame 2 to 5. Pseudo Code for determining the Search window size (SW) is as follows,

```
If (Ref == 1)
        For All Modes
                SW = Step2*
Else  // rest Reference frames
        If (mode == 7)
                SW = Step1
        Else // Mode 1 ~ 6
                SW = Minimum (Step 1, Step 2*)
```

**** Step 2 is used if the smaller blocks have the same motion vector.***

Results of the "Proposed Algorithm" column in Table VII show that the complexity reduction is very attractive. For "brideg_far" and "bridge_close", it is as low as 4% of the original complexity; while for larger motion sequences, like "mobile" and "bus", it is around 20%. The PSNR only drops 0.016 dB on average with 0.82% increase in bitrate. Fig. 5 shows that the lossless SAD Reuse algorithm which is adopted in JM 9.6 as an option can give a speed up of 2 - 4 times in motion estimation time as compared to the original JM 9.6 full search, while our proposed algorithm has a speed up ratio of 8 – 13. Fig. 6 and 7 compare the bitrate – PSNR graphs using the JM 9.6 full search with our proposed algorithm. Since the difference in PSNR is very small, the two curves overlap for nearly all parts. Table VIII compares our proposed algorithm with other recent papers on a similar study.

These results indicate that our approach outperforms all other algorithms in namely all aspects.

Algorithms	% Increase of bitrate	Drop in PSNR	Complex-ity %
Our Proposed algorithm	**0.82**	**0.016**	**11**
Huang *et. al* [11]			10~67
Chen *et. al* [13]	<2	<0.1	25
Hsiao, Lee and Chang [12]			25
Shen and Huang [7]		0.05	50

Table VIII Results compared with other papers

Fig. 5. Speed up ratio of proposed algorithm compared with SAD Reuse

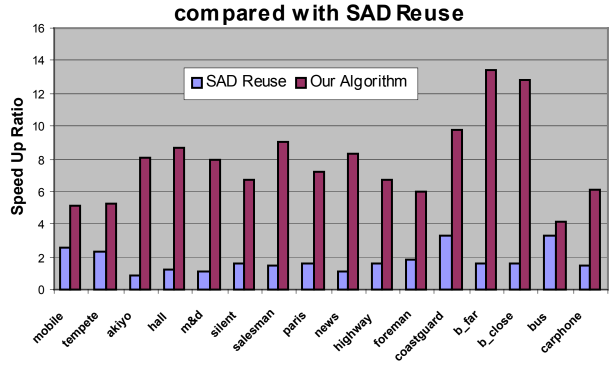


Fig. 6. Resut for "Mother and daughter"

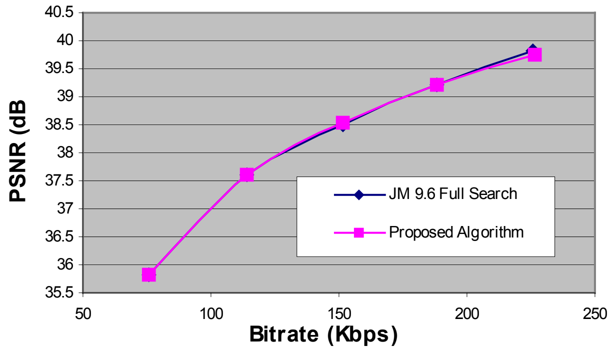
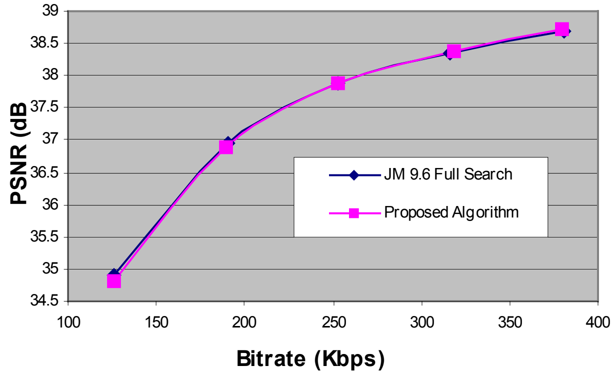


Fig.7. Resut for "Highway"



V. CONCLUSION

In this paper, we have proposed a simple and effective algorithm for multi-frame motion estimation. An initial analysis on the usage of various reference frames was made. We have found and verified by simulations that the motion vector obtained in the first reference frame is extremely useful for controlling adaptively the search window size of the rest of the reference frames. Besides, the motion estimation is firstly performed on smaller blocks, which in turn will benefit the search process for larger blocks if the smaller blocks have the same motion vector. By the refinement with a small search window around the common motion vector, the speed of motion estimation for large blocks can be improved. Experimental results show that our method can (i) save 90% of motion estimation computation, (ii) keep the video quality as good as the full search and outperforms other algorithms in the literature.

VI. REFERENCES

[1] Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)-Joint Committee Draft. Joint Video Team (JVT) of ISO/IEC MPEGand ITU-T VCEG, 7th Meeting: Pattaya, Thailand, 7-14 March, 2003.

[2] Yui-Lam Chan and Wan-Chi Siu,"An efficient search strategy for block motion estimation using image features". IEEE Transactions on Image Processing, Aug. 2001 Page(s): 1223 – 1238.

[3] Ko-Cheung Hui, Wan-Chi Siu and Yui-Lam Chan, "New adaptive partial distortion search using clustered pixel matching error characteristic", IEEE Transactions on Image Processing, May 2005 Page(s): 597 – 607.

[4] Al-Mualla M.E., Canagarajah C.N., Bull D.R., "Simplex minimization for single- and multiple-reference motion estimation", IEEE Transactions on Circuits and Systems for Video Technology Volume 11, Issue 12, Dec. 2001 Page(s):1209 – 1220.

[5] Chi-wang Ting, Wing-Hong Lam and Lai-Man Po, "Fast block-matching motion estimation by recent-biased search for multiple reference frames", ICIP 2004, Vol. 3 page(s): 1445 - 1448.

[6] Xiang Li, Eric Q. Li, and Yen-Kuang Chen, "Fast multi-frame motion estimation algorithm with adaptive search strategies in H.264", ICASSP 2004, Page(s):iii - 369-72.

[7] Yangfei Shen and Chao Huang, "Fast multi-frame motion estimation algorithm in H.264", ICSP 2004 proceedings.

[8] Hung-Ju Li, Ching-Ting Hsu and Mei-Juan Chen, "Fast multiple reference frames selection method for motion estimation in JVT/H.264", 2004 IEEE Asia-Pacific Conference on Circuits and Systems, Page(s): 605 - 608.

[9] Chi-Wang Ting, Lai-man Po and Chun-Ho Cheung, "Center-biased frame selection algorithms for fast multi-frame motion estimation in H.264", Proceedings of the 2003 International Conference on Neural Networks and Signal Processing, Page(s):1258 - 1261.

[10] Wu D., Pan F., Lim K.P., Wu S., Li Z.G., Lin X., Rahardja, S., and Ko C.C., "Fast intermode decision in H.264/AVC video coding", IEEE Transactions on Circuits and Systems for Video Technology, Volume 15, Issue 7, July 2005 Page(s): 953 – 958.

[11] Yu-Wen Huang, Bing-Yu Hsieh, Tu-Chih Wang, Shao-Yi Chien, Shyh-Yih Ma, Chun-Fu Shen, and Liang-Gee Chen, "Analysis and reduction of reference frames for motion estimation in MPEG-4 AVC/JVT/H.264" ", ICASSP 2003, vol.3 Page(s):III - 145-8.

[12] Yi-Hon Hsiao, Tein-Hsu Lee and Pao-Chi Chang, "Short/long-term motion vector prediction in multi-frame video coding system" Image Processing, 2004. ICIP '04. 2004 International Conference onVolume 3, 24-27.

[13] Mei-Juan Chen, Yi-Yen Chiang, Hung-Ju Li and Ming-Chieh Chi." Efficient multi-frame motion estimation algorithms for MPEG-4 AVC/JVT/H.264". 2004. ISCAS Vol.3 Page(s):III - 737-40.