CENTER-BIASED FRAME SELECTION ALGORITHMS FOR FAST MULTI-FRAME MOTION ESTIMATION IN H.264

Chi-Wang Ting [1], Lai-Man Po [1] and Chun-Ho Cheung [1,2]

cwting@ee.cityu.edu.hk, eelmpo@cityu.edu.hk and terence@ieee.org
[1] Department of Electronic Engineering, City University of Hong Kong, Hong Kong
[2] Department of Information Technology, Hong Kong Institute of Technology, Hong Kong

ABSTRACT

The new upcoming video coding standard, H.264, allows motion estimation performing on multiple reference frames. This new feature improves the prediction accuracy of inter-coding blocks significantly, but it is extremely computational intensive. Its reference software adopts a full search scheme. The complexity of multi-frame motion estimation increases linearly with the number of used reference frames. However, the distortion gain given by each reference frame varies with the motion content of the video sequence, and it is not efficient to search through all the candidate frames. In this paper, a novel center-biased frame selection method is proposed to speed up the multi-frame motion estimation process in H.264. We apply a center-biased frame selection path to identify the ultimate reference frame from all the candidates. Simulation results show that our proposed method can save about 77% computations constantly while keeping similar picture quality as compared to full search.

1. INTRODUCTION

H.264/MPEG-4 AVC [1] is the latest video coding standard developed by the Joint Video Team (JVT) which is formed with ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Motion Picture Experts Group (MPEG) in 2001.

This new standard significantly outperforms the existing video coding standards. It saves half of the bit-rate when compared with the H.263, and only uses about quarter of the bit-rate for the MPEG-2. In other words, we can have 2 to 4 times the video quality by renewing the current video codec while keeping the same bandwidth requirements. It was suggested in [2] that even a little improvement on the video compression efficiency (e.g. 10%), would reduce 20 times of current Internet backbone traffic. This means the new standard not only pursues higher performance, but also raises new opportunities for various bandwidth demanding video applications, especially for those mobile devices that have only limited bandwidth in the wireless network such as General Packet Radio Service (GPRS).

As the cost for processing power and memory is reduced, more heavy coding strategies and complex codec can be supported. The significant gain of compression efficiency in H.264 is at the expense of increased computation and complexity. For example, advanced intra/inter-prediction modes, tree-structured macroblock partitioning, quarter-pixel motion compensation and multiple reference frames motion estimation, all these features help increase the compression efficiency, but introduce tremendous loading to the system [1].

Format	Sequence Name
CIF (352x288, 80 frames)	Claire
	Miss America
	Sales
SIF (352x240, 80 frames)	Football
	Garden
	Tennis

Table 1. Image sequences used for analyses and simulations.

In a hybrid-coding [3] video encoder, most of the computation is spent on motion estimation. Hybrid-coding is the basis of all video coding standards, in which intra-frame coding and inter-frame coding are combined to reduce the spatial and temporal redundancy. Motion estimation (ME) and motion compensation (MC) are the two major techniques of inter-frame coding for video compression. A good prediction can substantially decrease the bit-rate. Nevertheless, these techniques are computationally intensive. In the case of exhaustive search of all candidate blocks, up to 80% computational power of an encoder is consumed by motion estimation [4]. In H.264, motion estimation is allowed searching on multiple reference frames to further reduce the temporal redundancy. Certainly, the computational load due to motion estimation will increase with the number of reference frames, and the cost of motion estimation will dominate the complexity of the video codec. This is absolutely a challenging problem to implement the codec without hardware aid, especially for mobile computing devices that have limited computing power. Obviously, it indicates an eager need for faster motion estimation strategy.

In this paper, we present a simple and effective method to reduce the computational cost due to multi-frame motion estimation without significant quality degradation. Instead of checking all the blocks on each reference frame, we only search on a center-biased path so that an ultimate frame can be selected for final search. In Section 2, we will analyze the motion vector probability (MVP) distribution in multiple reference frames. The results encourage the formation of our method. In Section 3, our algorithm will be described in detail. In Section 4, our simulation results will be shown with some theoretical analysis on the gain. Finally, a conclusion is given in Section 5.

Average MVP distributions (%)			
	Search window size		
Ref. frame	w=7	w=15	w=30
t-1	48.83	46.83	46.17
t-2	18.00	18.17	18.17
t-3	5.83	6.17	6.17
t-4	7.33	7.67	7.67
t-5	3.17	3.33	3.50
t-6	5.17	5.17	5.33
t-7	2.00	2.33	2.33
t-8	4.33	4.50	4.50
t-9	1.67	1.83	1.83
t-10	3.83	4.00	4.17
Total	100.00	100.00	100.00

Table 2. Average MVP distribution of six sequences on 10 reference frames with different search window sizes.

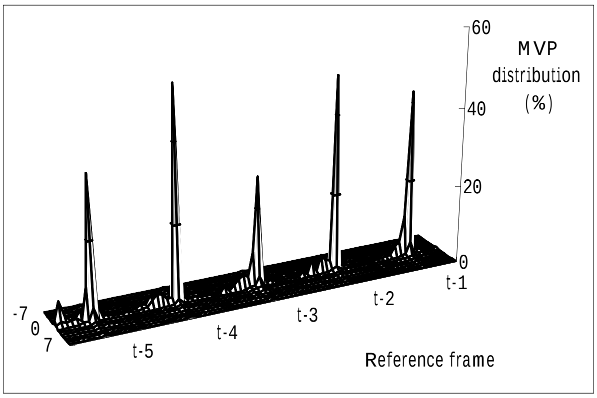


Fig. 1. Average local MVP distribution over search window size ±7 for six sequences in five reference frames.

2. ANALYSIS AND OBSERVATIONS

Many block matching motion estimation algorithms are inspired by the center-biased characteristic of motion vector distribution. New three-step search [5], four-step search [6], diamond search [7], and cross-diamond search [8] are some of the famous fast blocking matching algorithms (BMA) that utilize this characteristic. It reveals the fact that most real world sequences containing motions that can be located near the center of search window. As a result, these algorithms can substantially speed up the searching process more than 10 to 20 times. Therefore, it is interesting to analyze the motion vector probability distribution among multiple reference frames.

Our experiments are conducted with six CIF/SIF format images sequences shown in Table 1. Some video-conferencing-like sequences such as “Claire”, “Miss America”, and “Sales” contain smooth and gentle motions with static backgrounds, while the other 3 sequences consist of zooming, panning, and more vigorous motions. The image formats are CIF and SIF, and their dimensions are 352x288 and 352x240. 80 frames are analyzed with fixed 16x16 block size.

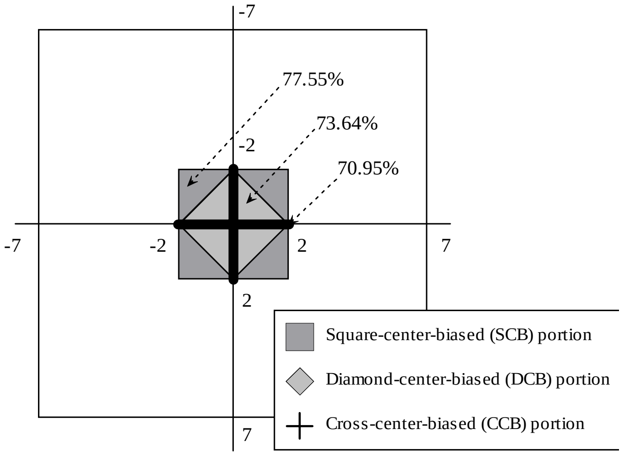


Fig. 2. MVP distribution accumulated on five reference frames with different center-biased characteristics.

First we analyze the impact of search window size on MVP distribution in CIF/SIF image sequences. The average distribution of the 6 sequences on 10 reference frames is shown in Table 2. Three window sizes, 7, 15 and 30 are tested. The result shows that increasing the window size tends to make more MVs distribute from F_{t-1} to other reference frames. There is about 0.2%-2.7% variation on MVP. It indicates that window sizes >7 virtually do not affect the MVP distribution for CIF/SIF sequences.

In order to explore the average motion vector probability distribution in frame-by-frame basis, the average local MVP found in each reference frame is plotted in Fig. 1. In this figure, the temporal distance does not influence the center-biased property. The higher the peak is, the more condensed the motion vectors are. This result shows that most motion vectors are still distributed around the center of the search window even multiple frames are referenced.

To further investigate the distribution characteristic, we classify the stationary blocks and quasi-stationary blocks (radius r = ±1 or ±2) into three regions, square-center-biased (SCB), diamond-center-biased (DCB) and cross-center-biased (CCB) as shown in Fig. 2. Among the five reference frames, 77.55%, 73.64% and 70.95% motion vectors are found in SCB, DCB and CCB regions respectively. Obviously, the largest square area contains highest amount of motion vectors. The statistic shows that CCB portion has highest compactness in terms of number of searching points, and it is effective to locate a large portion of motion vectors. This multi-frame distribution result is similar to that of single reference frame with about 4% loss in each portion.

From the experiment results, we also observe that motion vectors tends to be concentrated in the previous frame, F_{t-1}, for the sequences containing large and fast motions. This phenomenon is reasonable. In small motion sequences, the successive pictures are highly similar and correlated in temporal domain. In contrast, there is much lesser temporal correlation in large motion sequences, and it is supposed to decrease along with the temporal distance. As a result, multiple reference frames do not give significant improvement over single reference frame for large motion video.

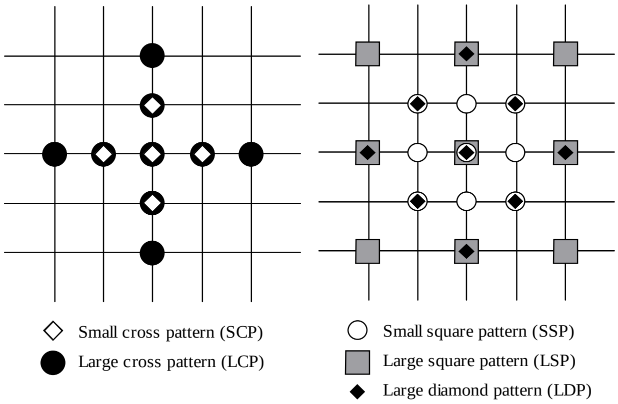


Fig. 3. Five center-biased pattern being analyzed in forming our proposed center-biased selection path.

3. PROPOSED SCHEME

In the design of H.264, the developers tend to improve the rate-distortion performance at all costs. A bundle of computation is input to the system for a little gain. For that reason, to design a fast algorithm, it is essential to keep the gain while trading it for the complexity.

In our analyses, we observed that the center-biased characteristic is preserved in multiple reference frames. There is a great tendency to find out the best-matched block from the region near to the search window center. Approximately 78% motion vectors can be found within the area of r = $\pm$ 2. Thus, we choose a center-biased selection path that can dominate over the other regions and in general give a relative high minimum hit rate of frame selection.

On the other hand, we find that cross-shaped patterns are more compact that they are favorable to cover a great portion of motion vectors with the least searching effort. In order to give maximum speed-up with minimum quality degradation, we draft six kinds of selection path in our fast motion estimation scheme. They are formed with the center-biased patterns shown in Fig. 3, except the center selection (CS) path. The pattern used by CS contains only one point in the center. The others are named according to the corresponding patterns: small-cross selection (SCS), large-cross selection (LCS), small-square selection (SSS), large-square selection (LSS) and large-diamond selection (LDS). Each selection path is the projection of its search pattern onto the reference frames. For example, in case *n* reference frames are allowed, our SCS path consists of *n* small-cross patterns in the center of the search window for each frame.

The first step of our strategy is to search through all the points in the selection path. The local minimum block distortion measure (BDM) found in the selection path is used as the indicator of the final reference frame. Here we assume that the global minimum occurs in the same frame as the local minimum of the selection path. This assumption follows the unimodal error surface model where the error is assumed to decrease monotonically towards the global minimum error, and it is commonly used by many BMAs. Lastly, we can apply any single frame ME methods to the selected frame in final search. For simplicity, we directly apply full search (FS) to the selected frame in our algorithm.

Sequences		SFS	CS	SCS	SSS	LCS	LDS	LSS
Claire	*	70.12	91.95	95.97	96.64	96.86	92.78	91.78
	**	70.12	86.58	94.36	95.36	95.81	89.03	88.10
MissA.	*	27.61	49.76	67.46	69.96	70.72	54.35	50.41
	**	27.61	25.46	51.42	55.06	57.22	34.90	31.64
Sales	*	17.46	96.76	98.29	98.83	98.59	96.74	95.95
	**	17.46	89.95	95.41	96.56	96.10	91.79	90.84
Garden	*	79.32	73.91	78.82	79.06	82.75	53.10	52.34
	**	79.32	3.44	39.52	39.86	57.59	21.85	21.63
Football	*	55.05	87.93	93.20	94.59	94.20	89.05	86.76
	**	55.05	65.09	76.61	80.22	79.48	71.57	68.84
Tennis	*	65.93	62.74	70.64	73.39	73.43	70.29	65.33
	**	65.93	24.03	33.56	39.93	39.50	36.34	33.03
Average	*	52.58	77.18	84.06	85.41	86.09	76.05	73.76
	**	52.58	49.09	65.15	67.83	70.95	57.58	55.68

*: Hit Rate (%), **: Minimum Hit Rate (%)

Table 3. Average hit rate and minimum hit rate w.r.t. MFS for 80 frames.

4. SIMULATION RESULTS

Our algorithm is simulated using the luminance component of the 6 sequences from frame 5 to 84 (total 80 frames). The maximum block displacement is set to $\pm$ 7 pixels and the number of allowed reference frame is set to 5. The block size is fixed at 16x16. The performances of multi-frame full search (MFS), single frame full search (SFS), CS, SCS, SSS, LCS, LDS, and LSS are compared. SFS functions as the baseline performance comparison for the others. The mean absolute error (MAE) is used as the BDM function.

In Table 3, the average hit rate and minimum hit rate for 80 frames are listed. The hit rate is defined as the percentage of successful detected frames with respect to MFS. The minimum hit rate is the lower bound of our detection successful rate. For those motion vectors fallen into our selection path, they must be able to hit as a result of FS. From the table, we can see that the performances of LCS and SSS are quite similar but LCS is generally more robust to different motions. LCS has the highest average hit rate (86.09%) of the 6 sequences. It always outperforms the SFS, and is almost the best for individual sequence. For small motion sequences like “Sales” and “Claire”, LCS has very high hit rates of 98.59% and 96.86% respectively.

To compare the picture quality, we measure the average MAE degradation per pixel with respect to MFS. The results are shown in Table 4. We can also find that LCS and SSS have similar performances in small motion sequences while LCS performs better in large motion sequences. LCS has the lowest average MAE degradation (0.187) of the 6 sequences. For small motion sequences like “Claire”, “MissA”, and “Sales”, LCS has extremely low degradation of 0.008, 0.037, and 0.011 respectively. The MAE per pixel is plotted against the frame number in Fig. 5(a) and 5(b). The performance of our algorithm is very close to MFS. In general, the degradations in small motion sequences are negligible. This algorithm is very suitable for video-conferencing applications.

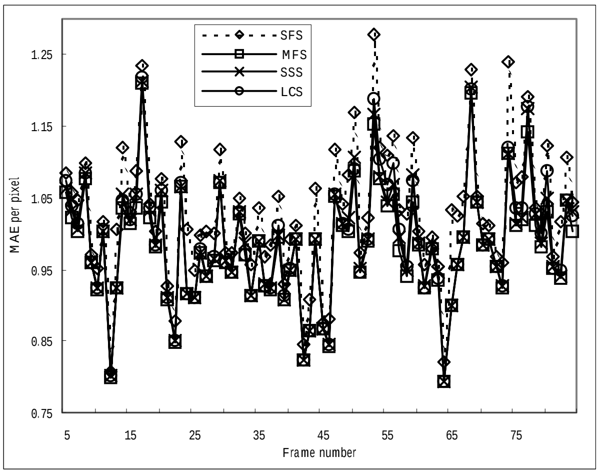
Table 5 shows the complexity reduction with respect to MFS. The theoretical values are calculated by the number of necessary searching points. The more the reference frames are allowed, the more the computations can be saved. Both LCS and

SSS can save about 77% computations constantly for 5 reference frames.

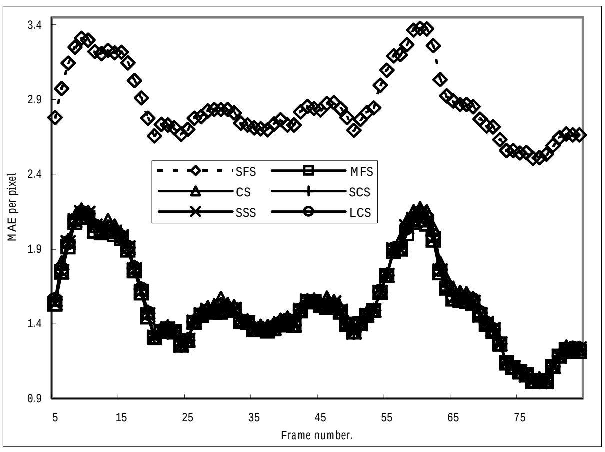
One can find that the hit rate performance does not always reflect the MAE performance. Although a relative high hit rate can be obtained in large motion sequences, the MAE performance is still unsatisfactory. This is because, for large motion sequences, the penalty of fail detection is large. The block distortion will be very large if a wrong frame is selected. This can be improved by selecting sub-optimal frames from the selection path, but trading for the searching speed.

5. CONCLUSION

In this paper, a novel frame selection method is proposed to speed up the multi-frame motion estimation in H.264. Based on the center-biased MVP distribution characteristic of real world sequences, we apply a center-biased frame selection path to efficiently locate an ultimate frame. Simulation results show that our algorithm using LCS has up to 99% hit rate. It can reduce about 77% computations while keeping the picture quality close to MFS. It is highly suitable for real-time video-conferencing applications.



(a)



(b)

Fig. 4. MAE per pixel against number of frame for (a) "Claire" and (b) "Sales" over 80 frames.

	MAE degradation per pixel						
Sequences	SFS	CS	SCS	SSS	LCS	LDS	LSS
Claire	0.045	0.028	0.010	0.006	0.008	0.026	0.040
MissA.	0.163	0.091	0.042	0.037	0.037	0.084	0.104
Sales	1.339	0.029	0.013	0.009	0.011	0.024	0.035
Garden	0.412	1.275	0.629	0.623	0.444	1.556	1.563
Football	0.378	0.454	0.265	0.211	0.176	0.273	0.334
Tennis	0.492	1.068	0.656	0.541	0.449	0.468	0.559
Average	0.471	0.491	0.269	0.238	0.187	0.405	0.439

Table 4. Average MAE degradation per pixel w.r.t. MFS for 80 frames.

	Complexity reduction (%)						
No. of ref. frame	SFS	CS	SCS	SSS	LCS	LDS	LSS
5	80.00	79.64	78.22	76.80	76.80	76.80	76.80
10	90.00	89.82	89.11	88.40	88.40	88.40	88.40
15	93.33	93.21	92.74	92.27	92.27	92.27	92.27

Table 5. The theoretical complexity reduction w.r.t. MFS.

6. ACKNOWLEDGMENTS

The work described in this paper was substantially supported by a grant from City University of Hong Kong, Hong Kong SAR, China. [Project No.7001385]

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)", Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, Mar. 2003.

[2] B. Girod and M. Flierl, "Multi-frame motion-compensated video compression for the digital set-top box", in Proc. IEEE ICIP, Sept. 2002.

[3] H. G. Musmann, P. Pirsh, and H. J. Grallert, "Advances in picture coding", Proc. IEEE, vol. 73, no. 4, pp. 523-548, Apr 1985.

[4] P. Pirsch, N. Demassieux, and W. Gehrke, "VLSI architectures for video compression – a survey", Proc. IEEE, vol. 83, no. 2, pp. 220-246, Feb 1995.

[5] 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.

[6] 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.

[7] 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, Aug 1998.

[8] C. H. Cheung and L. M. Po, "A novel small-cross-diamond search algorithm for fast video coding and videoconferencing applications", in Proc. IEEE ICIP, Sept. 2002.