One-Pass Computation-Aware Motion Estimation with Adaptive Search Strategy

Yu-Wen Huang, Chia-Lin Lee, Ching-Yeh Chen, and Liang-Gee Chen
DSP/IC Design Lab
Graduate Institute of Electronics Engineering and Department of Electrical Engineering
National Taiwan University, Taipei, Taiwan
Email: yuwen@video.ee.ntu.edu.tw

Abstract— **A computation-aware motion estimation algorithm is proposed in this paper. Its goal is to find the best block matching results in a computation-limited and computation-variant environment. Our new features are one-pass flow and adaptive search strategies. The prior scheme allocates more computation to the macroblock with the highest distortion in the entire frame step by step. This implies that random access of macroblocks is inevitable, and the search pattern must be determined in advance. The random access flow requires a huge size of memory for all macroblocks to store the up-to-date minimum distortions, best motion vectors, and searching steps. On the contrary, the one-pass flow can not only significantly reduce the memory size but also effectively use the context information of neighboring macroblocks to achieve faster convergence and better quality. Moreover, to improve video quality when computation resource is still sufficient, the search strategy is allowed to adaptively change from diamond search to three step search, and then to full search. Last but not least, traditional block matching speed-up methods are combined to provide much better computation-distortion curves.**

I. Introduction

Motion estimation (ME) is the heart of video encoders to remove temporal redundancy within video sequences. The block matching algorithm (BMA) is adopted by all of the existing video coding standards. Full search block matching algorithm (FSBMA) produces the best video quality but demands the most computation. Many fast BMAs, such as three step search (TSS) [1] and diamond search (DS) [2], have been proposed to speed up the FSBMA with acceptable loss of video quality or with sacrifice of simplicity and regularity.

Usually, ME is implemented with a hardware accelerator. The rapid improvements in processors and fast BMAs make the software encoder a feasible solution, too. However, when the encoder has to support a wide range of applications (e.g. QCIF (176×144) and CIF (352×288), 15 frames/s (fps) and 30fps), traditional BMAs will face two problems. First, a traditional BMA stops only when subsequent search points are all examined, and the searching process of a frame cannot be interrupted when the allowed time interval is passed, so real-time constraints may be violated. Second, once the BMA procedure is finished, it cannot be extended when extra computation is still available, so better video quality cannot be achieved.

Recently, the computation-aware (CA) concept is more and more important. In software implementations, processors may have to support video coding of different frame rates, frame sizes, and search ranges. In hardware implementations, even if the frame rate, frame size, and search range have been clearly determined, the computation resource (e.g. operating frequency) may still be adjusted according to the battery power for portable devices. The goal of CA BMAs is to find the best block matching results in a computation-limited and computation-variant environment.

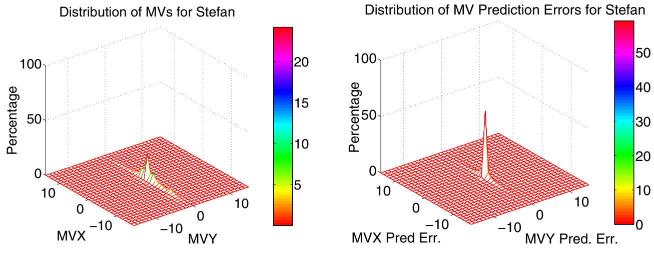
The authors of [3] are pioneers of CA BMAs. They contributed a novel scheme, which allocates more computation to the macroblocks (MBs) with the highest distortion in the entire frame step by step, as shown in the Fig. 2(d) of [3]. The main concept is that the larger the initial distortion, the more likely the distortion can be significantly reduced, and thus the more computation should be allocated. It is very simple and effective. Nevertheless, there are three problems in their scheme. First, random access of MBs is inevitable, requiring a huge size of memory for all MBs to store the up-to-date minimum distortions, best motion vectors (MVs), and searching steps. The advantage of MV predictors cannot be applied. For example, the predictive diamond search (PDS) [4] outperforms DS in both speed and quality. Second, the search pattern must be determined in advance. The advantage of adaptive search strategy cannot be applied, either. For instance, PDS is better in small motion cases, but TSS is better in large motion cases. The third problem is the poor hardware feasibility since it was intended for software. The distortion sorting operations can be easily implemented as hash tables or lists in software, but they are too expensive in hardware. The random access flow and enormous memory size are also harmful for hardware.

In this paper, a one-pass CA BMA with adaptive search strategy is presented. The ME is done MB by MB to solve the mentioned problems. The rest of this paper is organized as follows. In Section II, motion analysis is reported. In Section III, proposed algorithm is described. Simulation results are shown in Section IV. Finally, Section V gives a conclusion.

II. Motion Analysis

In this section, motion analysis is done in four aspects, as described in the following subsections. Four QCIF 30fps standard video sequences, Foreman, Silent, Stefan, and Weather, will be used in the statistics with search range as [-16,+15].

0-7803-8834-8/05/$20.00 ©2005 IEEE.



(a) (b)

Fig. 1. Statistics of motion for Stefan; (a) MVs; (b) MV prediction errors.

Foreman and Stefan are videos with large motion, while Silent and Weather are videos with small motion.

A. Motion Vector Predictor

MV predictors exploit the spatial correlation of neighboring MBs. Figure 1(a) and 1(b) show the distribution of MVs and that of MV prediction errors, respectively. FSBMA and the medium prediction from the left, top, and top right MBs are considered in the statistics. The distribution of MV prediction errors is much more concentrated around the origin than that of MVs, and the peak value at the origin increases from 24% to 59%. Starting from MV predictors makes PDS significantly better than DS in convergence speed and video quality.

Supplementary advantage of MV predictors is to support the rate-distortion optimized mode decision [5], known as Lagrangian method. Not only the distortion but also the MV costs are jointly considered in the mode decision. It is reported that 1dB PSNR gain can be achieved. However, in our experiments, we only use sum of absolute differences (SAD) as the matching criterion for generality because MV costs are dependent on entropy coding and quantization parameters.

B. Different Search Patterns

Different search patterns have different merits and thus should be combined into one CA BMA. Figure 2 compares FSBMA, TSS, and PDS. Among all frames, FSBMA gives the best quality (motion compensated PSNR). On average, PDS is better than TSS. However, when the camera pans very fast, TSS is better than PDS. The results are quite reasonable. When the motion field is small and regular, MV predictor works well, and the diamond pattern can quickly find a good match. As for TSS, the first step search points are dispersed, making final results tend to be trapped in local minima. On the contrary, when the motion field is large and complex, MV predictors do not work well, and the diamond pattern moves slowly toward the best MVs with a high probability of being trapped in local minima. In this case, TSS first glances the entire search area and has better chances to focus on the vicinity of global minimum.

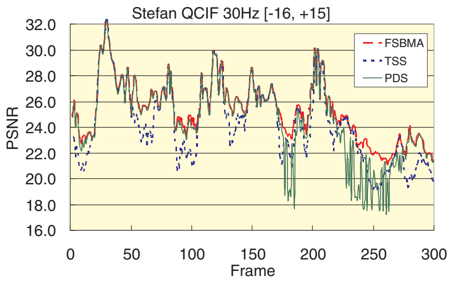


Fig. 2. Comparison of different search patterns.

TABLE I

PERCENTAGES OF IDENTICAL MVS BETWEEN PDS AND FSBMA.

Sequence	MVD≤0	MVD≤1	MVD≤2	MVD≤3
Foreman	97.41	94.17	79.84	80.27
Silent	99.51	97.66	92.24	91.30
Stefan	97.11	91.49	80.55	79.87
Weather	99.96	99.31	90.44	96.56

MV difference=MVD=|MVx-MVPx|+|MVy-MVPy|
MV=(MVx,MVy), MV predictor=MVP=(MVPx,MVPy)

C. PDS versus FSBMA

When the allocated computation for an MB has not been used up, a CA BMA will continue. However, if the global minimum distortion has been reached, searching more candidates is a waste. Therefore, there should be some detection to check if the optimal MV is reached for early termination of an MB. Thus, the saved computation can be utilized for later MBs. Table I lists the conditional probabilities of identical MVs between PDS and FSBMA. The smaller the distance from MV predictor to the final MV, the more likely the global distortion minimum is reached. Therefore, the MV differences (MVDs) defined in Table I can be used to skip BMA operations after PDS.

D. TSS versus FSBMA

Table II lists the conditional probabilities of identical MVs between TSS and FSBMA. After the first step search, if the best MV is the origin, it is very possible that the optimal MV will be found. Hence, the best MV right after the first step search can be used to stop the BMA operations after TSS.

E. Summary

The motion analysis is summarized as follows.

- MV predictors can achieve faster speed and better quality.
- PDS is suitable for small and regular motion fields.
- TSS is suitable for large and complex motion fields.
- PDS tends to reach the global minimum distortion when the MV predictor is close to the final MV.
- TSS tends to reach the global minimum distortion when the best MV of the first step is the origin.

TABLE II

PERCENTAGES OF IDENTICAL MVS BETWEEN TSS AND FSBMA.

Sequence	MV1st==0	MV1st!=0
Foreman	92.76	7.24
Silent	99.17	0.83
Stefan	93.97	6.03
Weather	99.72	0.28

MV1st: best MV after 1st step search

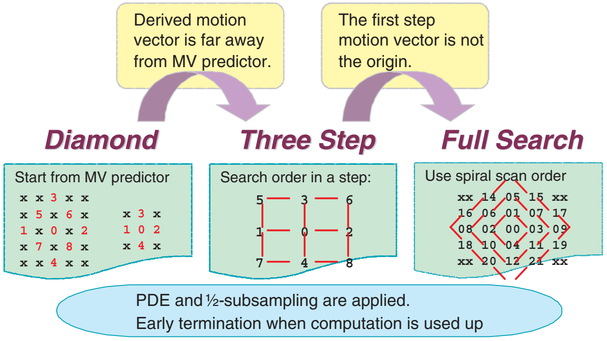


Fig. 3. Proposed adaptive search strategy.

III. Proposed Algorithm

In this section, our one-pass CA BMA will be proposed in four viewpoints as the following subsections.

A. Adaptive Search Strategy

Figure 3 illustrates our adaptive search strategy. First, PDS is selected as the initial search pattern for an MB. Second, when the PDS ends with available computation left for current MB, the search pattern is switched to TSS. Finally, FSBMA is adopted if TSS is finished with extra computation resource left.

In general, PDS is better than TSS in speed and quality, except for scenes with large and complex motion. In addition, CA DS and CA TSS performs better than CA FSBMA in the computation-distortion (C-D) plots, as stated in [3]. When the BMA is relatively abundant in computation resource, FSBMA still can improve the results. Based on the above reasons, we combine the three search strategies in this way.

As the analysis of Section II summarizes, detection of global minimum is employed. If the final MV of PDS is close to the MV predictor, TSS will not continue. If the best MV of the first step in TSS is the origin, FSBMA will not proceed.

B. Computation Allocation

In [3], the computation pool for the entire frame is determined with the constraints of video smoothness and the computation economy. However, for real-time bidirectional communication applications in which low latency is required, ME must be finished in time for every frame, and the frame computation pool must not exceed the reciprocal of frame rate (e.g. 1/15 sec for 15fps videos). In this paper, we focus on the MB-level computation allocation. The frame computation pool is taken as a given parameter.

Figure 4 is our computation allocation program. The new concept is dividing the computation resource into a base layer (BL) and an enhancement layer (EL). The BL guarantees the least computation for each MB. The EL allows each MB to receive additional computation according to the MB-level adjustment and early stop criteria. As shown in Fig. 4, the target search points per MB (MB_Tar_SPts) and that in BL (MB_Tar_SPts_BL) are user-defined. Afterwards, the frame target search points (FM_Tar_SPts) and that in BL (FM_Tar_SPts_BL) can be obtained from multiplying MB_Tar_SPts and MB_Tar_SPts_BL, respectively, with total number of MBs in one frame (TotalMB). The frame target search points in EL (FM_Tar_SPts_EL) is the result of subtracting FM_Tar_SPts_BL from FM_Tar_SPts.

- User definition
  - *MB_Tar_SPts*
  - *MB_Tar_SPts_BL*
- Frame level computation allocation
  - *FM_Tar_SPts = MB_Tar_SPts* TotalMB*
  - *FM_Tar_SPts_BL= MB_Tar_SPts_BL* TotalMB*
  - *FM_Tar_SPts_EL= FM_Tar_SPts- FM_Tar_SPts_BL*
- Macroblock level computation allocation
  - *AvgMinSAD= AccMinSAD/ DoneMB*
  - *MB_Alloc_SPts = MB_Tar_SPts_BL+ (Left_FM_Tar_SPts_EL/ LeftMB) * (InitSAD/ AvgMinSAD)*

Fig. 4. Proposed computation allocation.

- Frame layer computation allocation
- Initialize *AccMinSAD*, *DoneMB*, *LeftMB*, and *Left_FM_Tar_SPts_EL*
- Loop MBs
  - Initial block matching (MV=0) and find *InitSAD*
  - MB layer computation allocation
  - Block matching motion estimation
    - Adaptive search strategy (PDS, TSS, FSBMA)
    - Terminate when *MB_Actual_SPts >= MB_Alloc_SPts*
    - Terminate when quasi-optimal MV is found
  - Update *AccMinSAD*, *DoneMB*, *LeftMB*, and *Left_FM_Tar_SPts_EL*

Fig. 5. Macroblock procedure.

At the MB-level in Fig. 4, the concept of allocating more resource to MBs with larger distortions is adopted. The average minimum SAD of previous MBs (AvgMinSAD) is obtained as the accumulated minimum SAD (AccMinSAD) divided by the number of processed MBs (DoneMB). The allocated search points for an MB (MB_Alloc_SPts) is MB_Tar_SPts_BL plus the EL part, which is a product of two items. The first item denotes the future average search points per MB in EL, and is the left available computation pool of EL (Left_FM_Tar_SPts_EL) divided by the number of MBs that have not been processed (LeftMB). The second item denotes the ratio of initial distortion of current MB (InitSAD) to AvgMinSAD.

C. Macroblock Procedure

Figure 5 shows the macroblock procedure. The one-pass flow denotes that BMA is processed for MBs one at a time. Before entering the loop of MBs, frame level computation allocation and variable initialization are required. Inside the loop, the first step is to compute the SAD at the origin to find InitSAD for MB layer computation allocation. Then, adaptive search strategy determines the next search points. As long as the number of actual searched points reaches MB_Alloc_SPts, or the quasi-optimal MV (detection of global minimum distortion) is found, the BMA is terminated, and some variables are updated for the next MB.

D. Combination with Traditional Speed-up Methods

For each search point, partial distortion elimination (PDE) is applied to eliminate redundant SAD computation. Besides, 1/2-subsampling is also adopted.

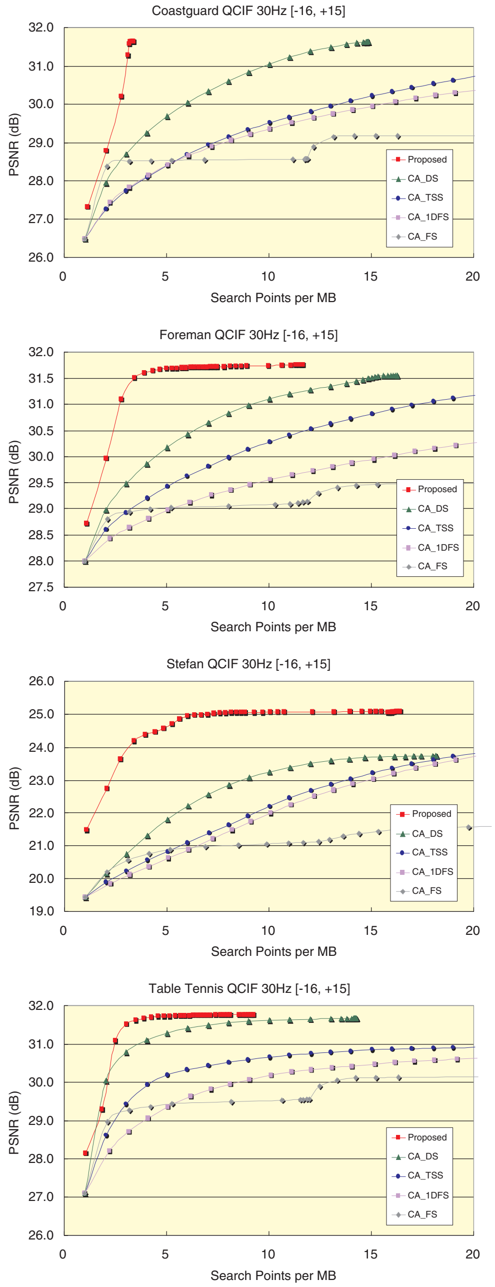


Fig. 6. Computation-distortion curves.

IV. Simulation Results

Figure 6 shows the C-D curves of the proposed algorithm and others in [3]. CA_DS, CA_TSS, CA_1DFS, and CA_FS are abbreviated from CA DS, CA TSS, CA one dimensional full search, and CA FSBMA, respectively. Many sequences were tested, but only Coastguard, Foreman, Stefan, and Table Tennis are shown due to the limited space and similar trends of C-D curves. The C-D performance of the proposed algorithm is significantly better than those of others. The average actually used computation of our algorithm cannot exceed a certain value for each sequence because our CA BMA early terminates when detecting that all MBs have reached the optimal MVs. Therefore, further increasing MB_Tar_SPts will not increase the actual search points. Furthermore, the best video quality of our CA BMA is only 0.1-0.2dB lower than that of CA_FS, and is better than those of remaining CA BMAs. However, this cannot be completely represented by Figure 6 because CA_FS reaches the best quality with many more search points.

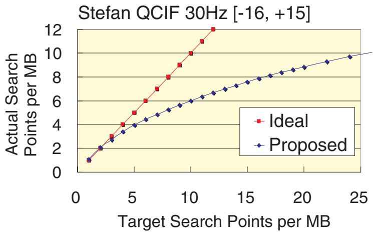


Fig. 7. Capability of proposed computation control.

Figure 7 shows the capability of the proposed computation control. The number of actual search points is never larger than that of target search points, which meets the real-time constraints. When the computation resource is little, the available computation will be exhausted. When the computation resource is abundant, the resource may not run out due to the detection of global minimum distortion.

In fact, if PDE and 1/2-subsampling are applied to [3], our algorithm cannot win so much, and even a small part of the CA_DS C-D curve may move to the upper left side of the proposed curve. The information of entire frame is indeed good for computation allocation. However, only our one-pass method can be benefited from Lagrangian mode decision, which enhances a lot of quality. Our strength also includes high hardware feasibility and much less memory requirement.

V. Conclusion

We presented a computation-aware motion estimation. The main idea is to convert the processing flow from random access to one-pass for hardware feasibility. Moreover, motion vector predictors and adaptive search strategy can thus be utilized for faster speed and better quality. Detection of global minimum distortion is also proposed to early stop the unnecessary computation. Simulation results show that the provided computation-distortion performance is relatively better.

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.*, 1981, pp. C9.6.1–C9.6.5.

[2] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block matching motion estimation," in *Proc. of IEEE Int. Conf. Image Processing (ICIP'97)*, 1997, pp. 292–296.

[3] P. L. Tsai, S. Y. Huang, C. T. Liu, and J. S. Wang, "Computation-aware scheme for software-based block motion estimation," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 13, no. 9, pp. 901–913, Sept. 2003.

[4] A. M. Tourapis, O. C. Au, and M. L. Liu, "Highly efficient predictive zonal algorithms for fast block-matching motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 10, pp. 934–947, Oct. 2002.

[5] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 688–703, July 2003.