A FRAMEWORK FOR FINE-GRANULAR COMPUTATIONAL-COMPLEXITY SCALABLE MOTION ESTIMATION

Zhi Yang [*], *Hua Cai* [†], *and Jiang Li* [†]

[*] College of Computer Science and Technology, Zhejiang University, Hang Zhou, China
[†] Media Communication Group, Microsoft Research Asia, Beijing, China

ABSTRACT

This paper presents a novel motion estimation (ME) framework that offers fine-granular computational-complexity scalability. In the proposed framework, the ME process is first partitioned into multiple search passes. A priority function is used to represent the distortion reduction efficiency of each pass. According to the predicted priority of each macroblock (MB), computational resources are then allocated effectively in a progressive way to achieve fine-granular computational-complexity scalability. Experiments show that our proposed scheme achieves progressively improved performance over a wide range of computational capabilities.

1. INTRODUCTION

With the rapid development of wired and wireless networks, more and more users are seeking real-time video communication services. However, real-time video coding faces a big challenge from computational complexity, especially for mobile devices such as Pocket PCs and handheld PCs, which are of weak computational capability and short battery lifetime. Because of the complexity constraint, many highly efficient but complex algorithms cannot be used directly for real-time video coding. Although one can simplify the algorithms to meet a specific scenario (e.g., a given video resolution and bit rate for a certain device), it is not a cost-effective way since there are so many different scenarios. Also, conventional encoders cannot adapt well to the varying computational requirements of video contents. Therefore, it is highly desirable to have a computational-complexity scalable video encoder that can offer a trade-off between coding efficiency and the embedded available computational performance.

In video coding systems motion estimation (ME) plays a key role in removing temporal correlation between video pictures. All the video standards that have so far emerged, such as H.263 and MPEG-1/2/4, are based on the ME in the encoding loop [1]. Meanwhile, ME is a very critical module of the encoder since it consumes most of the computing time. There are significant advances in fast ME techniques in recent years for alleviating the heavy computation load, such as the new three-step search (NTSS) [2], the diamond search (DS) [3], the circular zonal search (CZS) [4], and the predictive algorithm (PA) [5]. However, despite the significant speedups, ME still consumes the largest amount of computational resources, especially in real-time video encoding.

Computational-complexity scalable ME has been studied to further reduce the complexity of fast ME [6][7]. It also provides a proper trade-off between motion accuracy and time consumption such that it can adapt to the available computational resources dynamically. In Lengwehasatit's method [6], a partial-distance metric is used within the motion search process to eliminate unlikely candidates through a thresholding process that enables computation scalability. And in Mietens' method [7], complexity scalability is obtained by scaling the number of the processed motion-vector (MV) fields and the number of vector evaluations.

Different from previous works, in this paper, we present a novel ME framework that offers fine-granular computational-complexity scalability [1]. In the proposed framework, the ME process is first partitioned into multiple search passes. A priority function is used to represent the distortion reduction efficiency of each pass. According to the predicted priority of each MB, computational resources are then allocated effectively in a progressive way. As a result, the ME process can be stopped at any time with a progressively improved performance, and thus scalability is achievable. Furthermore, the proposed scheme can be easily integrated with many existing fast ME algorithms, such as NTSS [2], DS [3], CZS [4], and PA [5].

The rest of the paper is organized as follows. The new ME framework is presented in Section 2. In Section 3, the prediction of the priority for each search pass is discussed. Experimental results are shown in Section 4 and conclusions are drawn in Section 5.

2. PROPOSED ME FRAMEWORK

In the ME process, a recursive temporal prediction loop is employed to find the best MV that optimizes the rate-distortion performance. Usually, the prediction loop works as follows: for a given starting point, first check a number of candidate MVs and then select one MV from the candidates as the new starting point. The loop will repeat until either all of the candidate MVs have been checked or the stop condition is satisfied. Therefore, the ME process of a certain MB (say, the i^{th} MB) can be naturally partitioned into multiple search passes: $Pass_i(1)$, $Pass_i(2)$, ..., and so on. The j^{th} pass of the i^{th} MB, i.e. $Pass_i(j)$, would simply check $N_i(j)$ candidate MVs and determine the new starting point for $Pass_i(j+1)$. Fig. 1 illustrates an example of pass partitioning for the DS algorithm [3], where the ME process is partitioned into five passes, each having 1, 8, 5, 3, and 4 candidate MVs respectively.

After pass partitioning, the MV prediction of a certain MB is deployed in a progressive way. If all passes are searched, the encoder will get the best MV that is equivalent to that of the conven-

[*]The work presented in this paper was carried out in Microsoft Research Asia.

[1]To simplify terminology, the word *scalability* refers to *fine-granular computational-complexity scalability* hereinafter.

0-7803-8834-8/05/$20.00 ©2005 IEEE.

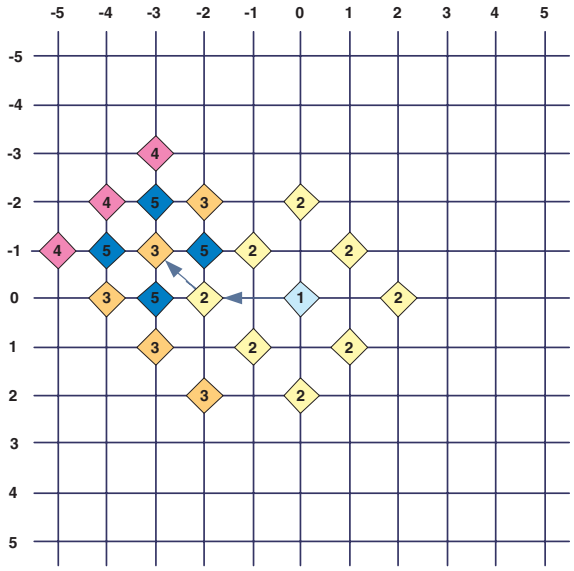

Figure 1: An example of pass partitioning for the DS algorithm. Numbered points are those candidate MVs that will be checked in the corresponding pass.

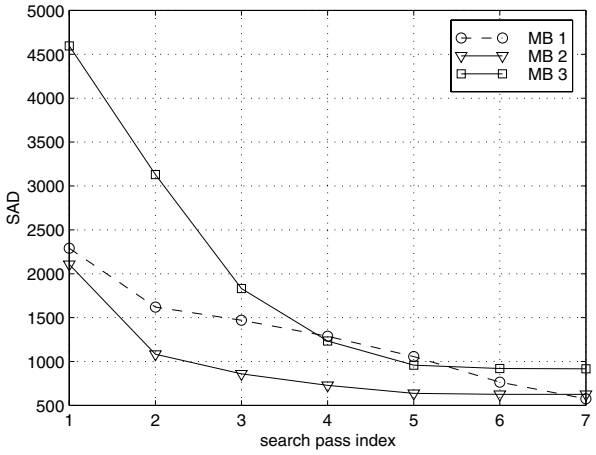

Figure 2: Distortion vs. number of passes (obtained with DS algorithm for the *Foreman* (CIF) sequence).

tional ME schemes. Meanwhile, the encoder also has the freedom to stop the MV prediction at any of these passes. Moreover, by selecting passes among a frame of MBs, scalability is achievable.

The simplest way to attain scalability is to select passes according to their indexes, that is, from the first pass of all MBs all the way down to the last pass of all MBs. As a result, all MBs have the same chance to refine their own MVs. However, its performance is not good since the search efficiencies of different passes, which are of the same pass index but from different MBs, might be quite different. Fig. 2 clearly demonstrates that different MBs might have quite different search efficiencies.

Instead of uniformly allocating computational resources to each MB, in our proposed framework, a more sophisticated approach is considered based on the priority of each pass. We measure the search priority of $Pass_i(j)$ by the reduced distortion per computing time:

$$P_i(j) = \frac{\Delta D_i(j)}{N_i(j)} = \frac{\Delta \text{SAD}_i(j)}{N_i(j)} \quad (1)$$

where $\Delta D_i(j)$ and $N_i(j)$ denote the distortion reduction and the number of checked MVs of $Pass_i(j)$ respectively. Particularly, we use the sum of absolute differences (SAD) in this paper to measure the distortion, and $\Delta\text{SAD}_i(j)$ is the reduced SAD of the i^{th} MB after performing $Pass_i(j)$. Furthermore, we use $N_i(j)$ to represent the consumed time since in most ME algorithms the computing time for each candidate MV is invariant (i.e., the distortion calculating is of constant complexity). Also note that only the predicted value of $P_i(j)$, denoted as $\hat{P}_i(j)$, can be obtained in practical systems since $\Delta D_i(j)$ cannot be calculated without finishing $Pass_i(j)$. Details of the priority prediction will be discussed in Section 3.

Now the computational resources can be allocated as follows: a priority table which contains N priority elements is first created for a video frame consisting of N MBs. Each priority element represents the predicted priority of the current unperformed pass for a certain MB. Next, the MB which has the largest priority is selected for one pass of ME. After finishing that pass, the new priority of its next pass is then predicted and the corresponding priority element is updated as well. This resource allocation process continues recursively until either the available computing time is consumed or there is no new pass left for any of the N MBs.

Ideally, the highest resource utilizing efficiency can be attained by the above method if the priority can be obtained accurately and the priority function for any MB is convex. However, the priority function is not always convex in practice. To make our algorithm more robust, we also take into account the non-convex case when predicting the priority.

On the other hand, it is critical to limit the implementation complexity for both priority prediction and table maintenance. In particular, we have observed that the most time consuming work for the table maintenance is to find the largest priority element from the table, especially for a large table. In one of our implementations, we separate the priority table into several sub-tables with different priority ranges, and all the MBs belonging to the largest prioritized sub-table are searched at every resource allocation step. Doing this can significantly reduce complexity, only with a slight performance loss.

3. PRIORITY PREDICTION

We have discussed a new ME framework in Section 2 that offers scalability. By smartly determining the priority of each ME search pass, an encoder can adapt to the limited/varying computational resources with efficient resource utilization. In this section, we present an effective and robust priority prediction method which does not require complex computation.

Usually, in fast ME algorithms, prior search passes are more likely to catch the optimal MV than the subsequent passes. This is the main reason that fast ME algorithms can dramatically reduce complexity. It also implies that the distortion reduction efficiency of a prior pass is usually greater than that of the subsequent ones. In other words, from the statistical point of view, the distortion reduction function for a ME process has a decreasing slope. Based

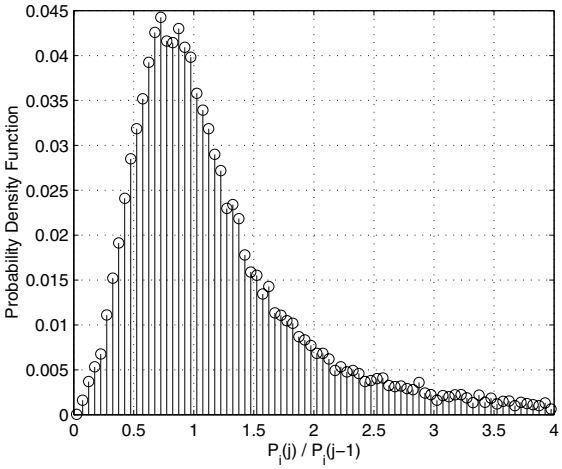

Figure 3: Probability density function of $P_i(j)/P_i(j-1)$.

on this assumption, the priority of $Pass_i(j)$ can be predicted as:

$$\hat{P}_i(j) = \begin{cases} \infty, & \text{if } j = 1 \\ \alpha \text{SAD}_i(j-1)/N_i(j), & \text{if } j = 2 \\ min\left(\beta P_i(j-1), \alpha \text{SAD}_i(j-1)/N_i(j)\right), & \text{else.} \end{cases} \tag{2}$$

In the above, only $\Delta D_i(j)$ needs to be predicted whereas $N_i(j)$ is explicit before performing $Pass_i(j)$. The priority is first set to infinity because we have no knowledge about each MB at the first pass of the ME process. Then, the predicted distortion reduction of the second pass is simply obtained from the current SAD scaled down by a pre-determined factor α. As for the remaining passes, priority $\hat{P}_i(j)$ is related to the actual priority of the previous pass, $P_i(j-1)$, scaled down by a pre-determined factor β. Meanwhile, a minimum value is chosen to ensure that $\hat{P}_i(j)$ is always within a reasonable range.

Note that in Eqn. (2) we assume the distortion reduction efficiency is convex for any ME process. However, it is not always convex in practice. Although the above method might still be robust for many non-convex cases where there is not any inefficient pass(es) (which has very small priority) before efficient pass(es), it cannot handle an inefficient pass which will block the search process. To make our algorithm more robust, we check the priority with another item:

$$\hat{P}_i^*(j) = max\left(\hat{P}_i(j),\ \frac{SAD_i(j-1)\cdot\gamma^K}{N_i(j)}\right) \tag{3}$$

where K denotes the number of consecutive inefficient passes prior to the current pass; and γ is a pre-determined scaling factor. From above equation, if consecutive inefficient passes are detected for a certain MB, it is believed that a global or near-global minimum is found, and thus low priority should be set for that MB.

From our testing, we observed that the performance is not very sensitive to the choice of parameters α, β, and γ. Hence we just set α and γ to $1/4$ and $1/8$ respectively throughout our experiments. As for β, its value can be calculated by averaging over the probability density function (PDF) of $H\left(\frac{P_i(j)}{P_i(j-1)}\right)$:

$$\beta = \int_0^{+\infty} x \times H(x)dx. \tag{4}$$

To improve robustness, only large $P_i(j)$ is used for collecting the PDF. Moreover, β can be either calculated beforehand or updated frame by frame. Fig. 3 shows one example of the PDF obtained from the *Foreman* (CIF) sequence. The corresponding β equals 0.95, which is used throughout our experiments. It can also be seen from Fig. 3 that the non-convex case (i.e. $P_i(j)/P_i(j-1) > 1$) happens very often.

4. EXPERIMENTAL RESULTS

Many experiments have been performed to evaluate extensively the performance of our proposed framework. The standard test sequences *Foreman*, *Carphone*, and *News* of CIF resolution at 30 fps are used as our test set. Only the first frame is encoded as an **I** frame and all others as **P** frames.

We implemented our framework upon the MPEG-4 encoder [8]. The DS algorithm [3] is used for ME with integer motion accuracy. The parameters α, β, and γ are invariant throughout the experiments. The priority table is split into 16 sub-tables as discussed in Section 2 in order to reduce the complexity of table maintenance. To better evaluate the proposed framework, two other schemes are also implemented as benchmarks for comparison. In the first benchmark (named *'Benchmark 1'*) the computational resources are uniformly allocated to different MBs. In the second benchmark (named *'Benchmark 2'*), we assume that the priority is known beforehand. It serves as a performance upper bound of our framework since the priority has to be predicted in practice (although it can only achieve sub-optimal performance due to the non-convex cases, we still treat it as the upper bound).

We first evaluate the resulting SAD for a random selected frame after checking a limited number of MVs. It is clear from Fig. 4 that, without considering the priority, the performance (achieved by 'Benchmark 1') is bad. On the contrary, our proposed framework achieves very good performance which is also close to the upper bound (i.e. 'Benchmark 2'). Similar performance is also observed from other frames.

We then evaluate the PSNR values of different sequences under a rate of 1024 kbps. As shown in Fig. 5, the simple 'Benchmark 1' scheme may suffer more than one dB loss compared with the performance upper bound. But our proposed framework significantly reduces the big gap, especially for the *News* sequence where our performance is very close to the upper bound. This demonstrates the effectiveness of our priority prediction approach used in the framework. It can also be seen that, by stopping MV prediction at any point on these curves, the encoder can easily adapt to the limited/varying computational resources only with a slight performance loss. This indicates that good embedded available computational performance is achievable.

5. CONCLUSIONS AND DISCUSSIONS

Computational-complexity scalability is an important yet practical topic in real-time video encoding. This paper presents a novel ME framework that offers fine-granular computational-complexity scalability. By partitioning the ME process into multiple search passes and prioritizing each pass according to its search efficiency, computational resources can be efficiently allocated in a progressive manner. Thus good embedded available computational performance is achievable. Good results have been observed in our experiments.

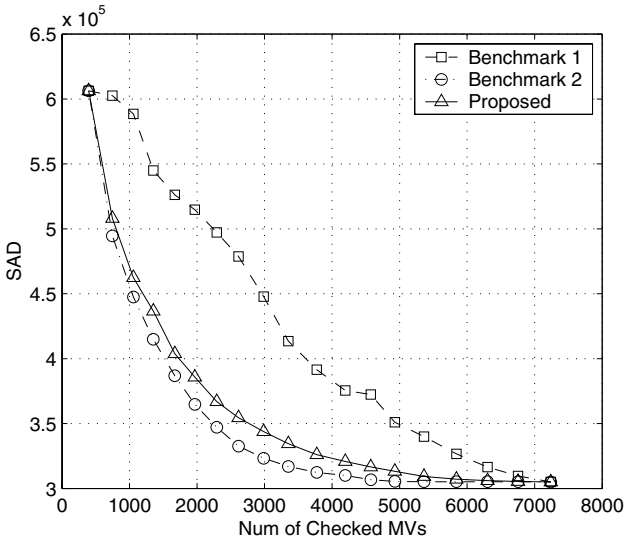

Figure 4: SAD vs. number of checked MVs (obtained from the 71^{th} frame of the *Foreman* sequence).

Many of the popular fast ME algorithms can be integrated with the proposed framework, as long as they can be partitioned into multiple search passes. Also, we believe that the performance of the new framework could still be improved by using a better priority prediction approach.

6. REFERENCES

[1] V. Bhaskaran and K. Konstantinides, *Image and video compression standards – algorithms and architectures*, Kluwer Academic Publishers, second edition, 1997.

[2] R. Li, B. Zeng, and M.L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 438–442, Aug. 1994.

[3] S. Zhu and K.K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 287–290, Feb. 2000.

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

[5] A. Chimienti, C. Ferraris, and D. Pau, "A complexity-bounded motion estimation algorithm," *IEEE Transactions on Image Processing*, vol. 11, no. 4, pp. 387–392, Apr. 2002.

[6] K. Lengwehasatit and A. Ortega, "Computationally scalable partial distance based fast search motion estimation," *Proc. ICIP'2000*, vol. 1, pp. 824–827, Sept. 2000.

[7] S. Mietens, P.H.N. de With, and C. Hentschel, "Computational-complexity scalable motion estimation for mobile MPEG encoding," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 281–291, Feb. 2004.

[8] Microsoft Corporation, "ISO/IEC 14496 (MPEG-4) Video Reference Software, Version 2.2.0," July 2000.

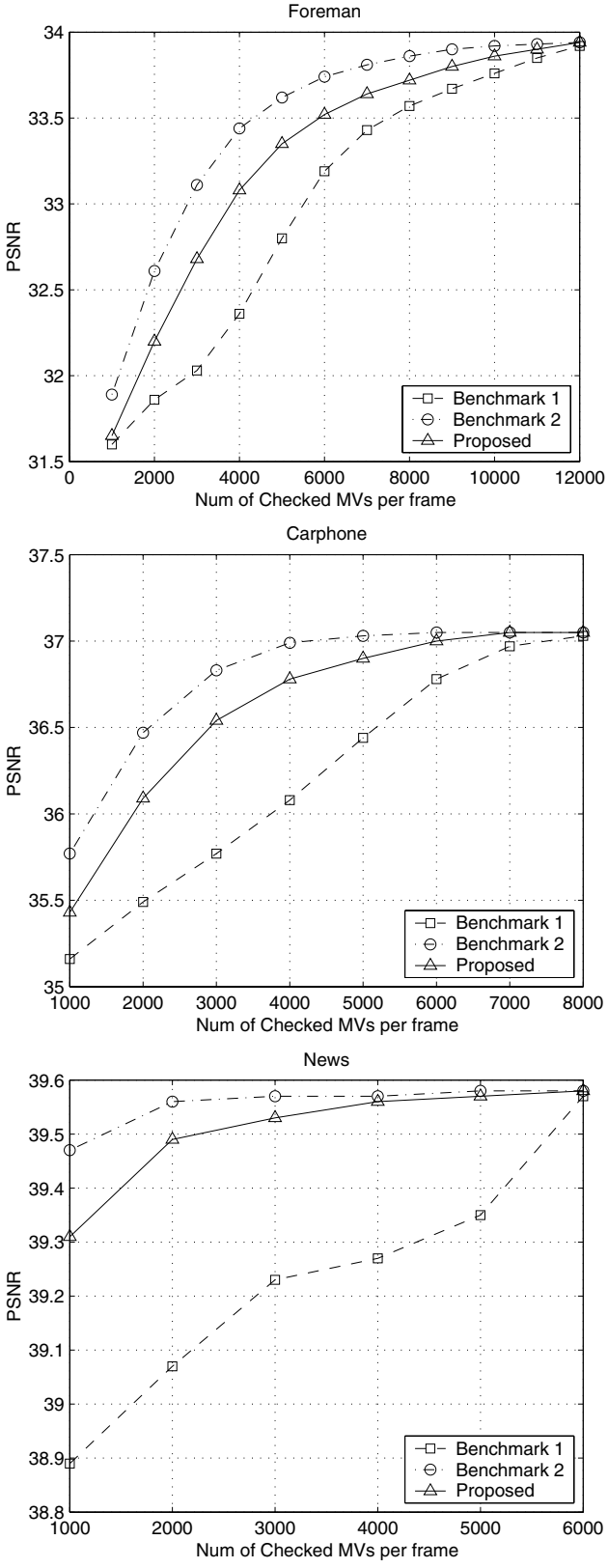

Figure 5: Comparative evaluation of the proposed framework.