SEARCH AREA SELECTIVE REUSE ALGORITHM IN MOTION ESTIMATION

Heejun Shim, Kyungsu Kang, Chong-Min Kyung

Korea Advanced Institute of Science and Technology
Department EECS
Daejeon, 305-701, Korea

ABSTRACT

Motion estimation has been widely studied and used to improve coding efficiency with small data access for power-saving. Conventional search area reuse algorithm requires small memory access by reuse of search area, but, suffers from coding efficiency degradation in fast motion video sequence. In this paper, we propose a search area selective reuse algorithm. The proposed algorithm well selects search center for reducing memory access in slow motion region and for tracking real motion in fast motion region according to the motion vector and search center information of the neighbor block. Compared to the conventional reuse algorithm, our experimental results show that the proposed algorithm prevents the coding efficiency degradation with small memory access overhead by tracking real motion in fast motion frames.

1. INTRODUCTION

In video coding system, motion estimation (ME) plays a key role to improve coding efficiency by reducing temporal redundancy within video sequence. ME takes more than 50% computational complexity in H.264/MPEG-4 AVC [1] which is the state-of-the-art video compression standard. Block matching algorithm, which is ME algorithm used in H.264/MPEG-4 AVC, is based on dividing a frame into macroblocks (MBs) (16x16 pixels), and searches for the best matching block with current MB within search area in a reference frame. Because ME process requires very high computational complexity, it has been implemented as a dedicated hardware to reduce its execution time at the cost of hardware area and power consumption.

In VLSI design, ME adopts an on-chip memory as an intermediate memory for power saving purpose [2]. Power reduction can be achieved by lower power consumption of smaller memories compared to frame memory at the cost of additional on-chip memory and data transfer. Some data reuse algorithms using on-chip memory were proposed. In [3], Wuytack showed the data reuse exploration and hierarchical memory mapping method. Brockmeyer analyzed the power consumption according to memory hierarchy [4]. Tuan defined the redundancy access factor and categorized the data reuse algorithm from level A to D [5]. Generally, level C reuse algorithm which reuses the overlapped searching region between two successive MBs in the horizontal direction is widely used in [6, 7, 8] for reasonable on-chip memory size and its power consumption. These data reuse algorithms are based on the characteristic of the center-biased MV distribution. Therefore, in fast motion video sequence with low frame rate, with small search range or with high resolution, they suffer from coding efficiency degradation due to low real motion tracking ability.

In this paper, we propose a search area selective reuse algorithm which decides whether search area is reused or not considering coding efficiency and the amount of memory access. Compared to the conventional reuse algorithm, the proposed algorithm shows the similar coding efficiency and memory access in slow motion region and compensates for coding efficiency degradation in fast motion region by tracking the real motion with small memory access overhead.

This paper is organized as follows. In section 2, the conventional search area reuse algorithm is described. The proposed selective reuse algorithm is described in section 3. In section 4, we show simulations and results. Finally, we summarize our results in section 5.

2. THE CONVENTIONAL REUSE ALGORITHM

In JM10.2 [9], which is a H.264 reference software, motion vector prediction (MVP) is used as search center in ME. MVP is defined in H.264 standard and should be calculated for encoding process. Generally, MVP is derived from motion vectors of neighbor blocks in the current frame by using median calculation. In the standard, one MB can be partitioned into some subblocks with block size from $16 \times 16$ to $4 \times 4$ where every subblock requires ME process. MVP of each subblock can be different from each other so that it leads to very large memory access for search area from frame memory. Therefore, for hardware implementation of ME, the fixed search area for one MB encoding is widely used with on-chip memory which is as large as the search area. By this method, search area once-fetched to on-chip memory is reused for ME process of all subblocks in a MB.
In addition, the conventional data reuse algorithm makes the search center of all MBs fixed to (0,0) as shown in Fig. 1(a). This is based on the center-biased MV distribution, i.e., the real motion vector is likely to be found near the same location of the current MB in a reference frame. Shown as a cross-hatched region in Fig. 1(a), the overlapped search area between two successive MBs in the horizontal direction can be reused in on-chip memory for the next MB encoding. However, if the size of real motion vector is larger than search range, the fixed search center of (0,0) results in coding efficiency degradation because ME cannot trace the real motion vector [10, 11]. To solve this problem, search range should be set large enough to include large motion vectors in fast motion video sequence. This method, however, requires large on-chip memory and more memory access. As shown in Fig. 1(b),(c), as the search range increases from 16 to 32, the required on-chip memory size is increased from \(48 \times 48\) to \(80 \times 80\) bytes and the required memory access for search area for a MB encoding is increased from \(48 \times 16\) to \(80 \times 16\) bytes.

3. SEARCH AREA SELECTIVE REUSE ALGORITHM

For real motion tracking and high data reuse with small on-chip memory size, we propose a search area selective reuse algorithm. Our algorithm is based on the high spatial correlation of search center of the previous MB and the real motion vector of the current MB. In addition, we assumed that motion vector is smoothly changed between two successive MBs. We adopted the fixed search area for one MB encoding and assumed that on-chip memory is as large as the search area as mentioned in the previous section.

In the proposed algorithm, the search center of the current MB is set to MVP for tracking the real motion in the fast motion region. In slow motion region, the search center is determined by shifting rightward by 16 pel the search area of the previous MB to make the overlapped region to be reused in on-chip memory, which means that the vector pointing to search center is the same as that of the previous MB. This is presented in Fig. 2(a), where Prev.SCV is the vector pointing to previous MB’s search center from the previous MB and MVP is the motion vector prediction of the current MB. The solid line presents the MVP of each MB and the dashed line does the Prev.SCV in Fig. 2. To decide whether the current MB is in a fast motion region or not, we used the difference between MVP and Prev.SCV which stands for the size of motion change in two successive MBs as shown in Fig. 2(b). The detailed condition to get the vector for current search center...
(Current SCV) is as follows.

\[ D(x) = |MV P(x) - Prev.SCV(x)| \]
\[ D(y) = |MV P(y) - Prev.SCV(y)| \]
\[ D = \max\{D(x), D(y)\} \]
\[ Current\ SCV = \begin{cases} 
Prev.SCV & \text{if } D \leq \text{Threshold} \\
MVP & \text{otherwise}
\end{cases} \]

Figure 2(c) shows the example of the search area selection in our algorithm. In the second MB (#2), the difference between Prev.SCV and MVP is small so that search center is set to Prev.SCV for search area reuse. In this case, the real motion is well found near the Prev.SCV because the selected search center is close to MVP. When the difference is large enough due to the accumulation of error between the real motion and the previous MB’s search center such as the third MB (#3), search center moves to MVP for real motion tracking.

Once search center is moved to MVP, following MBs such as the fourth MB and fifth MB in Fig. 2(c) are likely to reuse the search area sequentially with search center well tracking the real motion. Search center movement to MVP requires high memory access because on-chip memory should be all updated by data centered around the MVP in frame memory. So, we decided the threshold to \((7/8) \times \text{Search \ Range}\) based on a good trade-off between the amount of the memory access and the degradation of real motion tracking ability.

\section{4. SIMULATIONS AND RESULTS}

The proposed algorithm was simulated in JM10.2 [9] and compared with the conventional data reuse algorithm with same search range in terms of coding efficiency and the required memory access. We tested it on five fast motion video sequences: Canoe (30fps), Football (30fps), Foreman (15fps), Bus (10fps), and Stefan (10fps). H.264 baseline configuration file is used and only one reference frame is allowed. Search range is set to 16. Fractional ME is disabled for only integer ME algorithm comparison. RD-optimization is off and Hadamard transform is deactivated. Each sequence is encoded with only one I frame initially and followed by P frames (IPP...). All of the block partition types in H.264 are enabled. Memory access is theoretically calculated by summing the number of memory accesses for the current MB and its search area.

Three test algorithms are compared: 1) conventional data reuse algorithm (Conv.), 2) no data reuse algorithm (Org.) of which search center is always MVP for tracking the real motion at the cost of large memory access, and 3) our algorithm (Ours). In no data reuse algorithm (Org.), on-chip memory should be updated by data centered around the MVP at every MB encoding. Therefore, the required memory access of no data reuse algorithm is about 2.34 times that of the conventional reuse algorithm.

\begin{table}[h]
\centering
\caption{PSNR Gain and Memory Access Overhead compared to Conv. in Overall Frames and Fast Motion 10 Frames}
\begin{tabular}{|c|c|c|c|c|}
\hline
Video & Gain,Overhead (dB), (%) & Overall Frames & Fast Motion Frames \\
& & Ours & Org. & Ours & Org. \\
\hline
Canoe & 0.76, 11 & 0.82, 134 & 3.27, 8 & 3.35, 134 \\
Football & 0.43, 8 & 0.47, 134 & 0.91, 16 & 1.34, 134 \\
Foreman & 0.10, 3 & 0.12, 134 & 1.41, 13 & 1.46, 134 \\
Bus & 0.23, 8 & 0.32, 134 & 0.86, 9 & 1.07, 134 \\
Stefan & 0.06, 5 & 0.12, 134 & 0.46, 7 & 0.55, 134 \\
\hline
\end{tabular}
\end{table}

Fig. 3 shows rate-distortion curve of Foreman video sequence in overall frames and fast motion frames. The difference of rate-distortion curve in overall frames is small in Fig. 3(a). However, Foreman video sequence has a fast camera movement in the last part of the sequence. So, in this fast motion frames, PSNR can be significantly improved by our algorithm as shown in Fig. 3(b). The rate-distortion curve of ours is similar to that of no data reuse algorithm, which means that our algorithm well tracks the fast real motion. In the memory access results in overall frames, compared to no data reuse algorithm, our algorithm requires small memory access like the conventional data reuse algorithm. And, in the fast motion frames, our algorithm prevents the degradation of coding efficiency at the cost of small memory access overhead.

Table 1 presents the PSNR gain and memory access overhead compared to the conventional data reuse algorithm in various video sequences with bit rate of about 512kbps. This table shows that our algorithm achieves large PSNR improvement, which is similar to that of no data reuse algorithm, with small memory access overhead in the fast motion frames.

\section{5. CONCLUSION}

In this paper, search area selective reuse algorithm is proposed to prevent the coding efficiency degradation with small memory access overhead compared to the conventional data reuse algorithm. Our algorithm selects the search center considering high data reuse and real motion tracking according to the neighbor block’s motion vector and search center information. Compared to conventional reuse algorithm, the proposed algorithm compensates the coding efficiency degradation in the fast motion frames with small memory access overhead.

\section{6. REFERENCES}


