Reducing Computations in MPEG2 Video Decoder

V. Moshnyaga, and Kenji Wakisaka
Department of Electronics Engineering and Computer Science, Fukuoka University
8-19-1 Nanakuma, Jonan-ku, Fukuoka 814-0180, JAPAN

ABSTRACT

Abstract—This paper presents a new approach to reduce computations and memory accesses in MPEG video decoder. The approach is based on observation that existing decoders spent many redundant computations and memory accesses on processing macroblocks of bidirectionally predicted frames whose content remains actually same to adjacent reference frames. Experiments show that omitting such macroblocks from decoding can reduce the amount of computations and memory accesses required by the B-type frames almost by 43% on average without any impact on picture quality.

1. INTRODUCTION

1.1 Motivation

Emerging video application, such as portable DVD players, video phones, etc. have elevated needs for low-energy video decoding hardware. The standard video decoding scheme (Fig. 1) employed by MPEG assumes that each frame is structured as a set of macroblocks (each of N×N pixels in size), and the macroblocks are processed sequentially i.e. one after another. The tasks involved in the scheme are variable length decoder (VLD), inverse quantization (IQ), inverse DCT (IDCT), motion compensation (MC), etc. Traditionally, these tasks apply the same computational pattern to every block of the sequence even though the content of the block may not differ from the previous one or the variation is not distinguishable for the human eye. The details can be found in the standard, but the important facts for the paper are following: 1) the standard defines tree types of frames, called I-, P- and B- frames. These frames use zero, one or two reference frames for MC, respectively; 2) the reference frames are stored in the frame memory; 3) the static macroblocks, i.e. those, which have the same picture as the reference frame, are represented by zero motion vectors in the encoded video stream; 3) the I-frames in the stream are decoded through the VLD, IDCT and IQ, while P- and B-frames require MC; 4) Each B-type macroblock reconstruction requires N×N memory accesses to read the reference macroblock values, and N×N memory writes for storing the (4:2:0) chrominance values of the decoded B-type macroblock in the memory.

To store two reference pictures, the display picture and the VBV buffer in 4:2:0 chroma sampling and HDTV picture format, the MPEG2 decoders utilize external DRAM memory of 10-14MB in size [3-10]. Accessing the DRAM is significantly longer and takes more energy than arithmetic operations. Since the arithmetic operations are performed on registers, which have by orders of magnitude less switching capacitance than the DRAM, the most computationally intensive tasks, such as IDCT, are faster and less energy consuming than memory intensive frame reconstruction. As [1,2] show, the frame reconstruction alone takes almost 1/3 of the total decoding time and power. The reason is that it involves many computations and memory accesses. Therefore optimizing this function for less memory accesses is important.

The goal of the paper is to introduce a novel approach to lower the number of computations memory accesses in video decoder through macroblock data reuse.

1.2 Related Research

There are many approaches for reducing computational complexity and memory access count of video decoder. A considerable research effort has put on hardware solutions capable of lowering memory traffic, and increasing computational efficiency of decoding [3-10]. To achieve the goal, these hardware implementations extensively exploit memory access locality, memory distribution, partitioning, parallel processing, and data compression. Also a lot of work has been also done on algorithmic and architectural transformations for memory data reuse, memory access optimization, data allocation inside the RAM, and address/control optimization [2, 11]. Several algorithms have been proposed for optimizing computationally-demanding tasks such as IDCT, VLD, IQ [12,13]. We contribute to this research by an approach which dynamically eliminates redundant operations involved in decoding of B-frames. Although the I- and P-
frames are considered to be the largest in the GOP, the B frames require resource-intensive motion compensation operations and therefore frequently result in extensive decoding work [14].

In real video, the P and B frames are highly correlated. Hence, the amount of data which has to be processed depends on frame to frame image correlation. Fig. 2 shows a histogram of the number of image blocks that changed per frame in a sequence of MPEG2 video. We exploit the wide variation in this number to eliminate redundant decoding operations and memory accesses for the unchanged blocks.

1.3 Contribution

This paper proposes a new algorithmic approach for reusing block data in video decoding. Unlike existing techniques, the method adjusts the B-frame decoding operations and memory accesses to image variation statistics and by processing only those blocks whose content has been changed. Due to the applied mechanism of data-driven block characterization and selective block processing, the method can reduce the amount of computation/accesses by 30-40% on average, without affecting the picture quality. The scheme is simple yet efficient.

The paper is organized as follows. The next section presents the approach. Section 3 analyzes the performance. Section 4 summarizes the results and outlines our future work.

2. THE PROPOSED APPROACH

The technique we propose is based on observation that current decoding schemes do not fully exploit temporal correlation between the current picture at time instant $t$ and pictures prior to and after the time instant $t$. In MPEG2, MPEG4 standards, the decoding order and the displaying order are different, as shown in Fig.3. Namely, the “future” reference frame, $P_{1}$, that follows the B-type frames $B_{1}, B_{2}$ in the actual video stream, is decoded second, i.e. before $B_{1}$ and $B_{2}$. Similarly, decoding of the “future” reference frame, $P_{2}$, precedes decoding of the frames $B_{5}, B_{6}$, and so on. Furthermore, the motion of each macroblock in the “future” reference frame, $P(t+n)$, is compensated in regards to the “past” reference frame, $P(t-m)$. Because pictures in the same GOP are highly correlated, those macroblocks, $X_{i,j} \notin P(t+n)$, that are stationary, i.e. have zero motion vectors in the “future” frame, $P(t+n)$, are also stationary in those B-frames, which in reality take place in between $P(t-n)$ and $P(t+m)$. Since zero motion of these stationary macroblocks can be predicted before processing the B-frames, all the computations involved in decoding of these macroblocks are redundant.

We propose to detect macroblocks which exhibit zero motion in the “future” frame $P(t+m)$ in regards to the “past” reference frame $P(t-n)$ and omit these macroblocks from the B-frame decoding. Since the content of these macroblocks in P- and B-frames is the same, and the P-frame macroblocks are already in the memory, no reconstruction is needed. It means that neither MC computations nor memory reads/writes are required. The unchanged P-frame data are simply reused for display. The only price we pay for the reuse is a small memory for keeping $W \times H$ binary marks assigned to each macroblock of the reference frame.

Fig.4 outlines the flowchart of the proposed approach. The dashed line outlines steps for one B-frame. Notice that determining the stationary macroblocks does not require extra computations since it is performed during the motion prediction of the P-frame. During this step, we mark those macroblocks which have zero motion vectors by one, while the other macroblocks are marked by zero. When decoding the B-frames, positioned in between the reference frames $P(t-n)$ and $P(t+m-1)$, we check the marking of each macroblock $X_{i,j}$, and if it is set to one, we omit the decoding operation. Otherwise, the macroblock motion is decoded as usual.
3. EXPERIMENTAL RESULTS

The proposed approach has been tested in the modified MPEG2 video decoding environment on four standard video sequences: Miss America (176x144 pixels, 150 frames), Carphone (176x144, 382), Mobile (176x144, 398) and Salesman (252x288, 300). In each sequence, the GOP contained twelve frames; the distance between the past and the future references was two. In all sequences the macroblock size was 16x16 pixels and the search region was ±16 pixels. The performance of the proposed approach has been compared to the traditional (non-optimized) decoding scheme.

Fig. 5 illustrates the locations of macroblocks (denoted by asterisk) in the “Salesman” test sequence, frame 185, which were omitted from decoding by the proposed approach. As we see, 147 macroblocks out of 300 were skipped. We found that the number of skipped macroblocks strongly depends on picture correlation and varies as along tested sequences as well as along frames. Fig. 6 shows the number of macroblocks, skipped by conventional decoding scheme (Org) and the proposed scheme (prop) for each frame of the tested video sequences. We observe that the proposed approach shows very good results on slow-motion video sequences, such as Miss-America, Salesman and Carphone, skipping more than 90% macroblocks in a frame on a peak. The best results (almost 100%) are obtained for the frames 22-28, 33-38 of the Miss-America video sequence. The picture movement in these frames is almost none; so almost all frame macroblocks are skipped from the decoding.

Table 1 summarizes the results in terms of maximum and average ratio of skipped macroblocks to the total number of blocks in the frame, the reduction ratio (RR) achieved by the proposed approach in comparison to the original one (shown separately for all frames in the sequence and for the B-frames only), and the average picture error observed for the sequences. The error was calculated as:

<table>
<thead>
<tr>
<th>Video</th>
<th>Skipped % (Original) Max</th>
<th>Skipped % (proposed) Max</th>
<th>RR(%) all / B-only</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Salesman</td>
<td>90</td>
<td>67</td>
<td>96</td>
<td>72</td>
</tr>
<tr>
<td>Miss-A</td>
<td>89</td>
<td>38</td>
<td>99</td>
<td>65</td>
</tr>
<tr>
<td>Carphone</td>
<td>83</td>
<td>35</td>
<td>91</td>
<td>51</td>
</tr>
<tr>
<td>Mobile</td>
<td>13</td>
<td>4</td>
<td>16</td>
<td>9</td>
</tr>
</tbody>
</table>
Notice that the proposed approach affects picture quality very little (almost none) even when skipping 43% of the total number of B-type macroblocks on average (Miss America sequence). The measured picture error is less than 1% for all the sequences.

4. CONCLUSION

We have presented an approach to reduce encoding computations for B-type frames. As experiments showed, the proposed approach allows us to efficiently avoid unnecessary computations for 43% of the total number of macroblocks in video stream in comparison to traditional decoding scheme without a visible impact on picture quality. Work is under way to extend the approach to non-static macroblocks.

5. REFERENCES