A Quality/Energy Tradeoff Approach for IDCT Computation in MPEG-2 Video Decoding*

Russell Henning and Chaitali Chakrabarti
Department of Electrical Engineering
Arizona State University, Tempe, AZ 85287-5706
rhenning@asu.edu, chaitali@asu.edu

Abstract - It would be desirable, in terms of energy conservation, to use a low complexity approximate algorithm to do all IDCT computation in an MPEG-2 decoder. However, there is a significant quality penalty associated with this approach that may not always be acceptable. A practical algorithmic method is presented here for achieving quality/energy tradeoff for a hardware 2-D IDCT in an MPEG-2 decoder. By allowing multiple algorithms to run on the same IDCT hardware, multiple quality/energy tradeoff modes can be supported, and the quality that results when an approximate algorithm is used can be increased. For example, by simply using an approximate algorithm for all B frame IDCT processing, while using a conventional algorithm for I and P frames, it will be shown that 23% energy reduction can be achieved compared to using the conventional algorithm exclusively. This significant energy reduction requires a quality tradeoff of only 0.3 dB or less average PSNR.

Introduction

The future of hand-held devices that can receive video content will likely depend on finding effective energy management techniques for standards like the MPEG-2 video decoder. The Inverse Discrete Cosine Transform (IDCT) is the most computationally intensive portion of the MPEG-2 video decoder. Thus, it would be desirable, in terms of energy conservation, to use a low complexity approximate IDCT algorithm to do all IDCT computation in the MPEG-2 decoder. However, there is a significant quality penalty associated with such a design. In some cases, this quality penalty may be acceptable. For instance, it is probably more desirable to a mobile user to get more usage out of a battery than it is to receive the best quality. The level of quality degradation that is acceptable, however, depends on the user and the content being viewed. The user may even be in a situation where the device can be plugged into a power outlet and, naturally, expects the highest quality video available.

Dynamically varying an implementation based on nonstationary data characteristics so that the energy consumption is reduced has been studied in [1-3]. For example, Goel and Shanbhag showed that the energy consumption of a particular Reed-Solomon codec implementation can be reduced by 55% on average by powering down taps that are not required to meet a desired bit error rate for an input with dynamically varying SNR [2]. Lengwehasatit and Ortega have considered tradeoffs between quality and speed for DCT approximations [4], but not energy dissipation. They showed significant speed gains, around 25%, in software JPEG encoding by choosing from a variety of approximations, depending on the quantization resolution used for each 8X8 block.

In this paper, we present a practical algorithmic method for achieving quality/energy tradeoff for a hardware 2-D IDCT in an MPEG-2 decoder. Exact and approximate algorithms are allowed to run on the same hardware. As a result, the same IDCT hardware can provide multiple quality/energy tradeoff modes. In addition, the quality that results when an approximate algorithm is used can be significantly increased with a small energy penalty. The exact and approximate algorithms are mixed at the inter frame level in a thoughtful manner to achieve different amounts of quality/energy tradeoff. We will show, for example, that by simply using an approximate algorithm for all B frame IDCT processing, while using a conventional algorithm for I and P frames, 23% energy reduction can be achieved

* This work has been supported by the NSF-S/IUCRC Center for Low Power Electronics.

compared to using the conventional algorithm exclusively. This significant energy reduction requires a quality tradeoff of only 0.3 dB or less average PSNR. It will also be shown that practical quality/energy scalability can be achieved by employing multiple mixing methods at steps of 6-14% energy reduction with only 0.3-0.7 dB average PSNR degradation. We will see that mixing exact and approximate algorithms in this manner for fixed bitrate decoding actually does not increase noticeable quality variance between frames. Thus, using such an approach, very high quality can be maintained while significantly reducing energy consumption of IDCT computation.

Quality/Energy Tradeoff Approach

Our approach to trading off IDCT quality for energy consumption in MPEG-2 video decoding involves mixing exact and approximate algorithms in a thoughtful manner. Two exact and two approximate 8 point 1-D IDCT algorithms have been chosen to study the effectiveness of this approach. The 1-D IDCT algorithms can be used to compute the Row-Column 2-D IDCT. A nice feature about the Row-Column 2-D IDCT algorithm, is it allows a number of different 1-D IDCT algorithms to be extended in the same manner to perform 2-D computation. Other 2-D algorithms tend to be quite different from one another and relatively complicated. Thus, with a Row-Column implementation, a variety of algorithms can more easily share the same hardware, a key to our quality/energy tradeoff approach.

The theoretical minimum for the number of rational multiplications required in an exact 1-D IDCT algorithm has been found to be 11 [5]. As will be shown in the next section, multiplications are the most energy consuming operations in a hardware implementation of an IDCT. Thus, exact 1-D IDCT algorithms that achieve this minimum are of great interest for low energy implementation. For this study, the 11 multiplication, 29 addition Chen-Wang (CW) 1-D IDCT algorithm [6] implemented in the MSSG MPEG-2 decoder [7] has been chosen.

Another important exact 1-D IDCT algorithm can be implemented with only 5 multiplications and 29 additions [8]. We call this method the Scaled Exact (SE) 1-D IDCT algorithm, because the inputs to the IDCT algorithm are scaled. The algorithm actually requires 13 multiplications per 1-D IDCT, however the 8 scaling multiplications can be combined with the inverse quantization calculations of the MPEG-2 decoder, so that no additional multiplications are actually required. A drawback of this algorithm is that though it is an exact algorithm, for the same finite word length, its quality is slightly lower than the 11 multiplication CW algorithm.

To compute a 1-D IDCT with fewer than 5 multiplications, approximations can be employed. For this study, we have derived two multiplierless 8 point, 1-D IDCT approximations from the exact 1-D IDCT algorithm in [9]. Similar approximations can be found in [4]. The difference between our two approximations is accuracy, which translates to differences in the number of additions required. These approximations use scaled inputs similar to the SE algorithm, so we call the more accurate 1-D IDCT approximation Scaled Approximate 1 (SA1) and the less accurate approximation Scaled Approximate 2 (SA2). SA1 requires 34 additions per 1-D IDCT, whereas SA2 only requires 28. Figure 1 gives pseudo-code for these approximate algorithms. The X's are IDCT inputs, while the x's are IDCT outputs. The if-else statement at the beginning of the pseudo-code indicates the coefficients used by SA1 and SA2, the only difference between the approximations. Multiplications by these coefficients are implemented with shifts and adds.

It would be most desirable, in terms of energy conservation, to use the lowest compexity algorithm we have described, SA2, to do all IDCT computation in the MPEG-2 decoder. However, there is a significant quality penalty associated with this that may not always be acceptable. By allowing multiple algorithms to run on the same IDCT

```
if(SA == 1)              /* first stage */                /* second stage */   /* third stage */
 {                       a0 = X0;                         tmp1 = a0+a1;        x0 = b0+b7;
 C5  = 0.375;            a1 = X1;                         tmp2 = a0-a1;        x1 = b1+b6;
 C7  = 0.25;             a2 = C5*X2-X3;                   b0 = tmp1+a3;        x2 = b2+b5;
 C9  = 0.5;              a3 = X2+C5*X3;                   b1 = tmp2+a2;        x3 = b3+b4;
 C10 = 0.875;            a4 = C10*X4+C7*X5-X6-C9*X7;      b2 = tmp2-a2;        x4 = b3-b4;
 }                       a5 = C7*X4+C9*X5+C10*X6-X7;      b3 = tmp1-a3;        x5 = b2-b5;
else if(SA == 2)         a6 = -X4+C10*X5-C9*X6-C7*X7;     b4 = a4;             x6 = b1-b6;
 {                       a7 = C9*X4+X5+C7*X6+C10*X7;      b5 = a5;             x7 = b0-b7;
 C5  = 0.5;                                               b6 = a6;
 C7  = 0.25;                                              b7 = a7;
 C9  = 0.5;
 C10 = 1;
 }
```

Figure 1: Pseudo-code for the two Scaled Approximate 1-D IDCT algorithms.

hardware, multiple quality/energy tradeoff modes can be supported, and the quality that results when an approximate algorithm is used can be increased.

Two methods of mixing exact and approximate algorithms to increase quality while enjoying the energy conservation benefits of the approximate algorithms are proposed here. The first is to use an exact algorithm for I and P frames of a group of pictures (GOP) and an approximate algorithm for B frames. This method is based on two characteristics of a GOP. First, B frames are not referenced when differential encoding is applied to 8X8 blocks in P and B frames. Thus, quality reduction associated with approximate processing is confined to B frames, rather than spread to other frames in a GOP. Second, many blocks of B frames are encoded differentially, so only the differential portion of the block is distorted by approximate IDCT computation. These blocks tend to be more robust to IDCT quality degradation. In a typical GOP, like one of the form IBBPBBPBBPBBPBB, this method allows the lower energy approximate algorithm to be used for 2/3 of the frames in a sequence, while the exact algorithm is used for only 1/3.

The second mixing method uses an exact algorithm only for the I frames, while an approximate method is used for P and B frames. Since an I frame can be the basis for further differential coding throughout a GOP, it is most important to preserve its integrity. Like the B frames, many of the blocks of a P frame tend to be differentially encoded. These blocks tend to be more robust to approximation than intra coded blocks, especially for earlier frames in the GOP. Thus, the approximate algorithm would be used for 14/15 of the frames in the example GOP and the exact algorithm would only be used for 1/15. These two mixing methods represent two extremes. Other methods, like using exact processing for a subset of P frames in the GOP are also possible.

One might intuitively expect using an exact algorithm for the I and P frames or I frame only and an approximation for the remaining frames to cause a large variance in quality between the exact coded frames and the approximate frames that might be noticeable while viewing the sequence. However, as will be shown in the Simulation Results section, this is not the case for decoding fixed bitrate encoded sequences with these approximations. Before quality of the algorithms and mixing methods identified in this section are discussed in more detail, the differences between them in terms of energy consumption will be estimated in the next section.

Energy Comparisons

Differences in the number and type of operations performed by each of the IDCT computation methods described in the previous section can lead to significant energy consumption differences when implemented. The greatest benefit occurs when these methods are implemented in hardware rather than software, since software typically requires overhead that dwarfs data-path operation energy. Actual energy consumption of each algorithm could vary

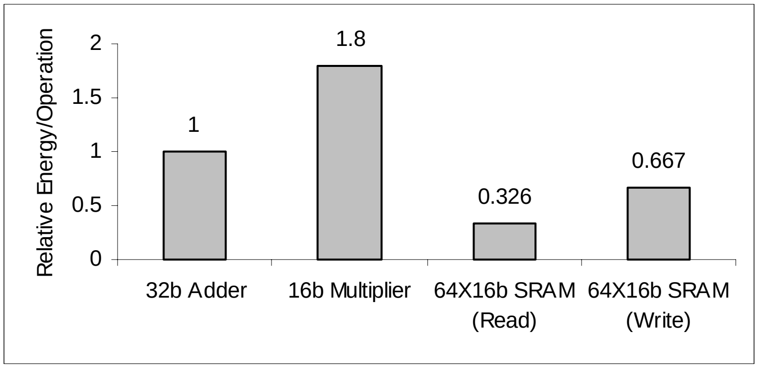

Figure 2: Relative energy/operation values used for relative energy estimates.

widely with architecture and technology used for hardware implementation. However, relative energy estimates allow practical energy comparisons with reasonable assumptions about architecture and technology.

In a good implementation, the three most significant contributors to the energy consumption of a hardware Row-Column IDCT are multipliers, adders, and the transposition memory. Other, contributors like control logic, clocks, multiplexing, and general purpose registers can be lumped into control. Though these algorithms also require a number of binary shifts, it is assumed that the shifts can be hard wired so they have essentially no effect on overall energy consumption. Depending on the architecture, some of these hard wired shifts may need to be multiplexed, which could increase control energy consumption slightly.

The energy consumption of a single multiplication, addition, or transposition memory access depends on word length. We assume that only one type of multiplier, adder, and memory will be used; though multiple copies of adders and multipliers may be included to increase throughput. The CW, SE, SA1, and SA2 algorithms were found to produce quality near that of double precision versions of each algorithm with 16 bit integer multipliers, 32 bit integer adders, and a 64 word SRAM using 16 bit words. The SA1 and SA2 algorithms do not actually require such large adders. However, since the goal is to be able to implement these approximations with an architecture that can also support one of the exact algorithms, 32 bit adders will be used.

Relative energy per operation estimates for multiplication, addition, and memory accesses can be found by assigning a certain functional unit a normalized energy per operation value of 1, a 32 bit addition in this case. Other operations are then assigned energy per operation values relative to the 32 bit addition. These values, given in Figure 2, are based on general relationships published in [10] but reflect scaling of energy with respect to adder word length and memory size. In [10], relationships are actually given for a 16 bit adder. Since 32 bit adders are of interest here, it is assumed that the energy per operation of the 32 bit adder is twice that of a 16 bit adder. The energy per operation relationships for SRAM reads and writes are given for a memory size of 1024 16-bit words in [10]. Here, the memory size of interest is 64 16-bit words, so the cost of a read or write is assumed to be 14.8% that of the 1024 word memory, based on the energy model in [11].

The relative energy per operation values in Figure 2 give rise to the relative energy estimates in Figures 3 and 4 for the various IDCT computation methods. To obtain these values, the number of each type of operation for a 15 frame GOP of the form IBBPBBPBBPBBPBB is simply multiplied by its energy per operation value. These intermediate values, as well as a control energy estimate, are added together and then normalized by the energy of the case where

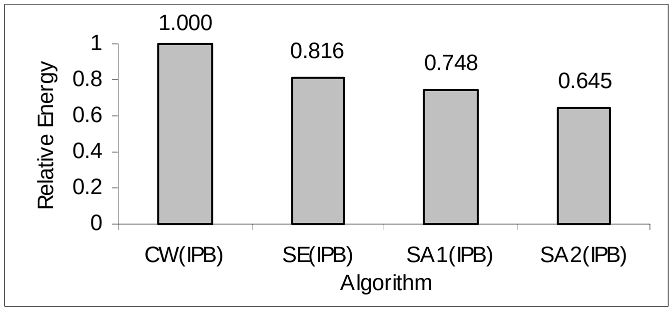

Figure 3: Relative IDCT energy estimates for a 15 frame GOP using the same algorithm for each frame.

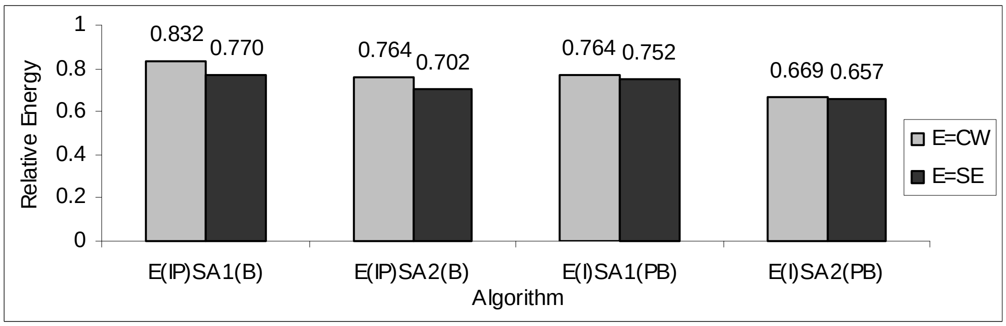

Figure 4: Relative IDCT energy estimates for a 15 frame GOP of form IBBPBBPBBPBBPBB using different algorithm mixing methods. Relative energy of 1 is for a 15 frame GOP using CW(IPB).

only the CW algorithm is used for the entire GOP. This case consumes the most energy of all the methods considered. The relative control energy estimate is obtained by assuming it is 10% of this highest energy case and remains constant for all algorithms. Figure 3 gives relative IDCT energy estimates for the 15 frame GOP when a single algorithm, either CW, SE, SA1, or SA2, is used for all frames of the GOP. The notation used in this figure indicates that the same algorithm is used by each frame type by including I, P, and B in parentheses after the algorithm name, e.g. CW(IPB). As can be seen from this figure, the SE algorithm consumes 18.4% less energy than the CW because it uses six less multiplications per 1-D IDCT. The SA1 algorithm which uses no multiplications but five more additions per 1-D IDCT than the exact algorithms consumes 8.3% less energy than the SE algorithm. The SA2 algorithm consumes 13.8% less energy than the SA1 algorithm with one less addition per 1-D IDCT than the exact algorithms and no multiplications.

Relative energy estimates for approaches that combine the exact algorithms with the approximate algorithms for the same 15 frame GOP are given in Figure 4. This figure uses the same relative scale as Figure 3, with a relative energy of 1 being equivalent to that of the case where the CW algorithm is used for all frames of the GOP. The notation used in this figure indicates which frame type uses which algorithm. For example, SA1(B) indicates that B frames employ the SA1 algorithm for IDCT computation, whereas E(IP) indicates that I and P frame IDCTs are computed with an exact algorithm.

Figure 4 directly compares employing the CW versus SE algorithm for the exact IDCTs in these approaches. By using an approximation for the 10 B frames of the 15 frames in the GOP, as is done in the first two cases of the figure, energy consumption is significantly reduced compared to using only an exact algorithm. However, by using the SE algorithm in these approaches, rather than the CW, significant energy can be conserved, nearly 8% in both of these cases.

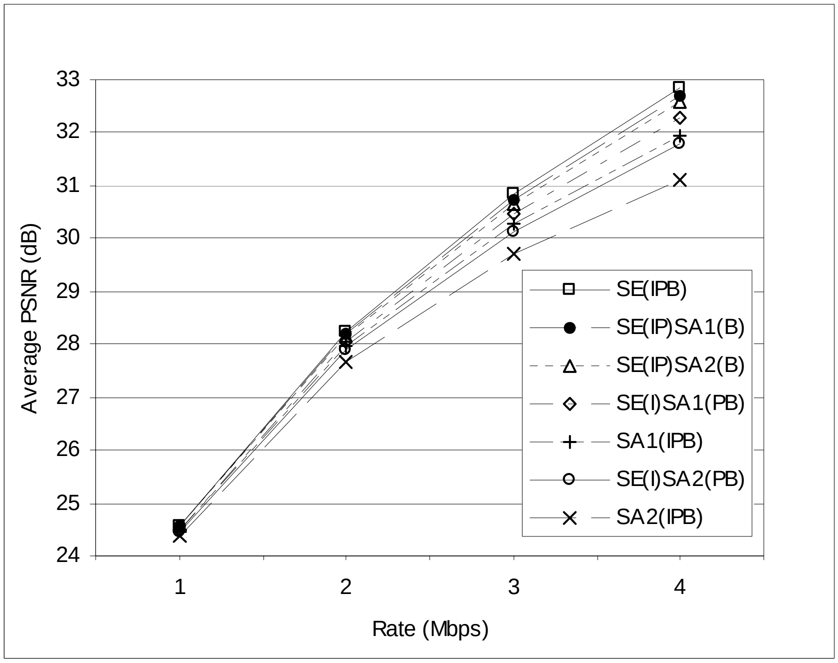

Figure 5: Rate-distortion curves for the Flower Garden sequence encoded with a 15 frame GOP of form IBBPBBPBBPBBPBB.

The third and fourth cases shown in Figure 4 use approximate algorithms for the 14 P and B frames of the 15 frame GOP rather than only 10. These approaches provide additional energy savings compared to the first two cases, though the savings are more significant when the CW algorithm is used for four fewer frames than when the SE algorithm is. There is little difference in energy consumption between using the CW algorithm and SE algorithm in the third and fourth cases because these exact algorithms are only used for one of the 15 frames in the GOP.

An interesting conclusion that can be drawn from these relative energy estimates is that the only algorithm using SA1 that can compete in terms of energy consumption with any of those using SA2 is SA1(IPB). However, SA1(IPB) only consumes lower energy than the CW(IP)SA2(B) algorithm. As will be shown in the next section, the quality of approaches using the SA2 approximation is so good, that using the SA1 algorithm at all is probably never practical.

Simulation Results

To distinguish the proposed IDCT computation methods in terms of quality, decoding simulations were run for four standard video sequences, Flower Garden, Football, Mobile, and Table Tennis, using the MSSG MPEG-2 decoder [7]. Each sequence has a frame resolution of 352X240 pixels, color subsampling of 4:2:0, and frame rate of 30 frames per second. This resolution was chosen because it is expected that the primary application of these methods would be in hand-held devices. Each sequence was encoded using the MSSG MPEG-2 encoder with a 15 frame GOP structure of the form IBBPBBPBBPBBPBB. Four fixed bitrates were employed for encoding, 4 Mbps, 3 Mbps, 2 Mbps, and 1 Mbps. Quantitative quality results were obtained in terms of PSNR measurements. The authors made subjective visual observations of quality and verified that they closely follow the quantitative results.

Figure 5 shows resulting rate-distortion curves for the Flower Garden sequence. The other three sequences exhibited similar results. The algorithms using the CW IDCT are not shown in this figure because their quality is only slightly better than the corresponding algorithms using the SE IDCT. Consider first the 4 Mbps average PSNR values.

The SE(IPB), SE(IP)SA1(B), and SE(IP)SA2(B) algorithms are all within 0.3 dB of each other. The SE(I)SA1(PB) result is only about 0.3 dB below the SE(IP)SA2(B) result. The SA1(IPB) and SE(I)SA2(PB) results are within 0.2 dB of each other, and SE(I)SA2(PB) is only about 0.5 dB below the SE(I)SA1(PB) result. The SA2(IPB) value is about 0.7 dB below that of the SE(I)SA2(PB) algorithm.

Taking these quality results and the relative energy estimates into consideration, a good choice for implementing an quality/energy scalable IDCT for the 4Mbps bitrate would be to enable decoding with the SE(IPB), SE(IP)SA2(B), and SE(I)SA2(PB) algorithms. Such an implementation would allow very high quality decoding, almost that of CW(IPB), when executing the SE(IPB) algorithm but at only about 18% the energy of CW(IPB). Nearly as high quality decoding is achieved with the SE(IP)SA2(B) IDCT algorithm but at significantly lower energy, about 14% less than the SE(IPB) algorithm. This is actually quite a remarkable result, because by mixing IDCT algorithms in this manner, the SA2 algorithm can significantly reduce IDCT energy dissipation while giving up nearly insignificant quality. Using SA2(IPB) gives considerably poorer quality than SE(IPB), about 1.7 dB for the Flower Garden sequence, while reducing energy by an additional 8% compared to SE(IP)SA2(B). Near the same energy reduction, 6%, can be achieved by mixing algorithms in SE(I)SA2(PB), but with only a 1.1 dB decrease in quality for the Flower Garden sequence. Thus, SE(I)SA2(PB) is probably a better alternative than the SA2(IPB) in terms of the tradeoff between energy and quality. Note that the energy differences between the methods chosen for scaling here would be even more dramatic if the CW algorithm were used instead of SE. For example, 23% energy reduction can be achieved using CW(IP)SA2(B) compared to using CW(IPB). This energy reduction requires a quality tradeoff of only 0.3 dB. However, recall that absolute energy consumption is significantly higher for CW(IP)SA2(B) than SE(IP)SA2(B).

As can be seen from Figure 5, all curves converge as the bitrate decreases. Note, however, that energy dissipation for each algorithm remains the same regardless of bitrate. Since the quality of 3 Mbps output is already significantly degraded compared to the 4 Mbps case, the 0.2 dB difference between SE(IPB) and SE(IP)SA2(B) is likely insignificant. Therefore, the only two algorithms that provide significant quality/energy scalability for 3 Mbps bitstream decoding are the SE(IP)SA2(B) and SE(I)SA2(PB) algorithms. These two algorithms are separated by about 0.5 dB in quality for the Flower Garden sequence and 6% in energy consumption. For 2 Mbps decoding, SE(I)SA2(PB) is probably always the best choice for decoding, since it produces quality only about 0.4 dB below SE(IPB). All algorithms are within 0.2 dB average PSNR for 1 Mbps, so the lowest energy algorithm, SA2(IPB) is always the best choice in this case.

For more detail about how the algorithms of Figure 5 achieve their average quality, PSNR results for consecutive frames of the sequence have been studied. PSNR results for the first 46 frames of the Flower Garden sequence decoded from the 4 Mbps bitstream are shown in Figure 6 for SE(IPB), SE(IP)SA2(B), and SA2(IPB). Lower bitrate decoding exhibits similar curves with proportionately less significant PSNR differences between the algorithms. In this plot, it is possible to see that not only are the I and P frames significantly better in SE(IP)SA2(B) compared to SA2(IPB), the quality of the B frames are remarkably improved to near that of the SE(IPB) case. This is because the I and P frames, which are decoded with the exact IDCT, are used as differential coding references.

A similar plot is shown in Figure 7 that compares the SE(I)SA2(PB) algorithm to SA2(IPB) and SE(IPB). The SE(I)SA2(PB) algorithm converges toward the quality of SA2(IPB) over the 15 frame GOP until it is increased once again for the last two B frames of the GOP. This increase in quality occurs because the next I frame, which is decoded with the higher quality SE algorithm, is used as a reference for some differentially coded blocks. The SE(I)SA2(PB)

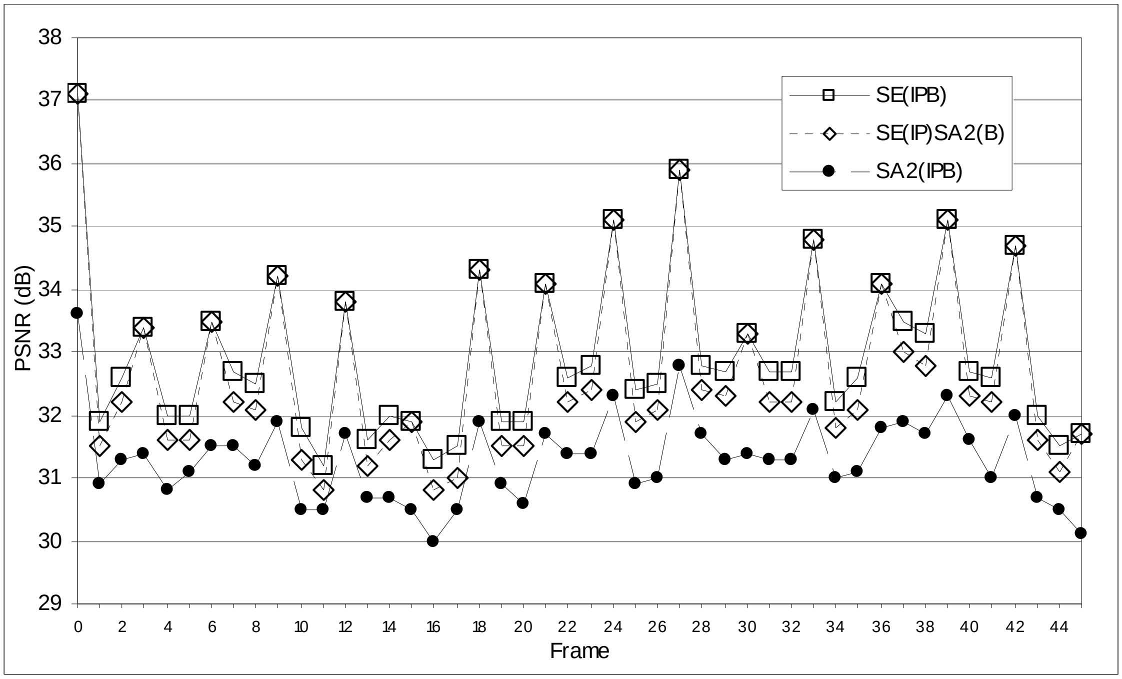

Figure 6: SE(IP)SA2(B) quality comparison for the first 46 frames of the Flower Garden sequence decoded from the 4 Mbps bitstream.

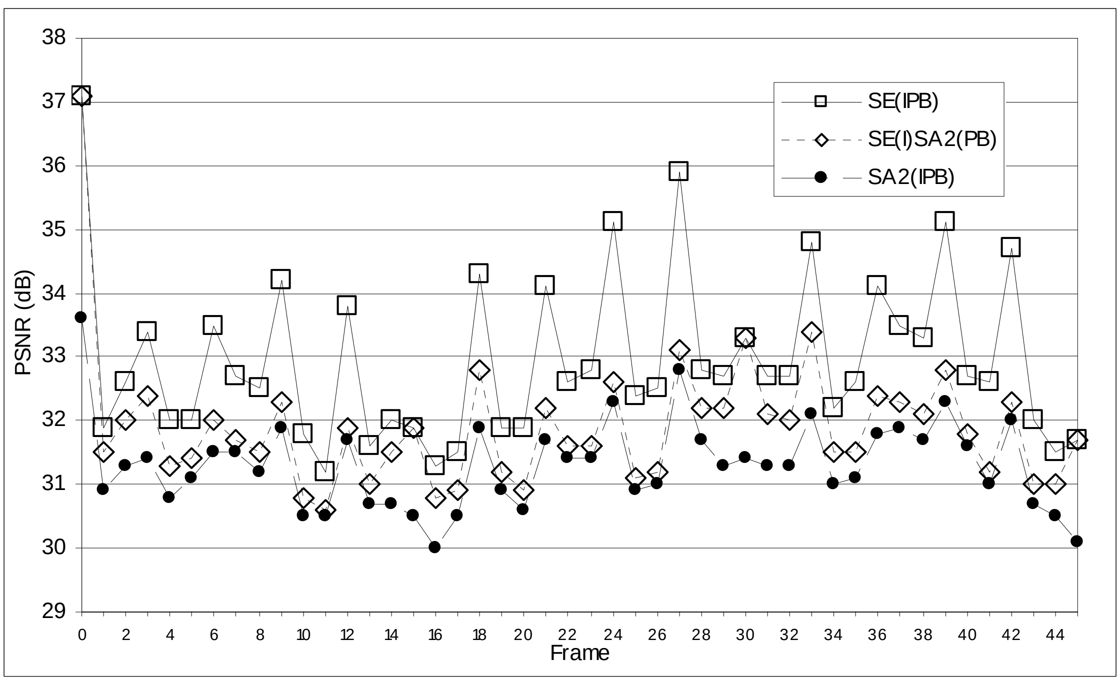

Figure 7: SE(I)SA2(PB) quality comparison for the first 46 frames of the Flower Garden sequence decoded from the 4 Mbps bitstream.

method has the side benefit, as does SA2(IPB), of reducing the difference between P frames and B frames compared to SE(IPB) to provide a more uniform quality video sequence from frame to frame of a fixed rate bitstream.

As mentioned previously, one might intuitively expect using an exact algorithm for the I and P frames or I frame only and an approximation for the remaining frames to cause a large variance in quality between the exact coded frames and the approximate frames that might be noticeable while viewing the sequence. This is not the case for decoding

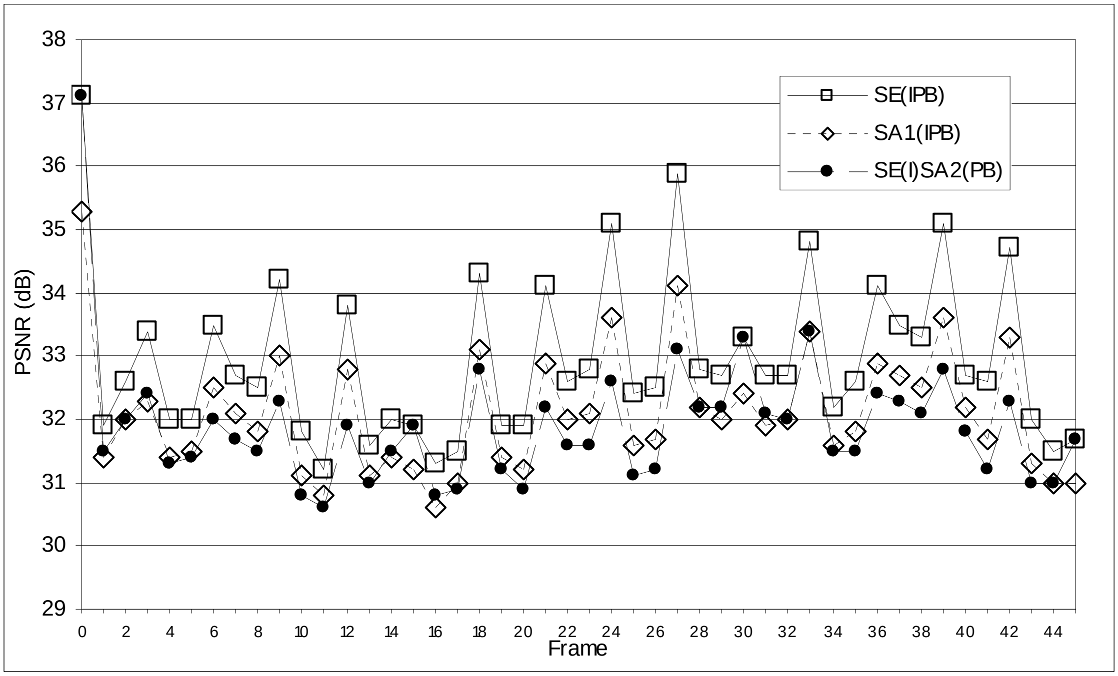

Figure 8: SA1(IPB) and SE(IP)SA2(B) quality comparison for the first 46 frames of the Flower Garden sequence decoded from the 4 Mbps bitstream.

fixed bitrate encoded sequences with these approximations. In the case where I and P frames are decoded exactly, this is due to a combination of the approximations being fairly high quality and the fact that overall B frame quality is raised near that of the exact algorithms because many blocks where the approximate IDCT is applied are encoded differentially using exact coded blocks as references. The difference between B frames using exact and approximate IDCTs becomes even smaller for lower bitrates.

In the case where only the I frame is encoded exactly, the quality of P and B frames is significantly lowered. However quality variance between frames is also lowered. For fixed rate encoding, P frames tend to have much higher quality than other frames when all frames are decoded using exact IDCTs. Since P frames are decoded with approximate IDCTs in this case, the quality of P frames is lowered, so quality is more uniform from frame to frame.

Another interesting comparison is the difference between SE(I)SA2(PB) and SA1(IPB). Though they appear to have similar quality from the average PSNR results of Figure 5, the difference in quality varies from frame to frame, as can be seen in Figure 8. SE(I)SA2(PB) outperforms SA1(IPB) near I frames in the sequence, and SA1(IPB) gives superior results near the middle of the GOP. From subjective visual analysis, it was verified that, overall, the quality of both methods are very similar. However, as seen in the previous section, the SA1 algorithm can't compete with the energy reduction capabilities of the SA2 algorithm.

Conclusions

The methods presented here show the great potential data-dependent algorithm mixing techniques have for applications requiring quality/energy scalability or reduced energy computation at the expense of a small amount of quality. Quality is actually reduced very little for the approximations tested. Other approximations and algorithm mixing methods may provide better and wider ranging quality/energy tradeoffs, and are a subject of our future work.

Possible schemes include only using the exact algorithms for the I frame and a subset of the P frames in a GOP. Based on the quality achieved, it may also be practical to mix exact and approximate algorithms at the intra frame level, using approximate algorithms only to decode non intra coded blocks, for instance. In addition, other methods that can be combined with our algorithmic technique to enhance its potential are being considered. Differences between algorithms in terms of word length can be exploited to reduce energy with smaller multipliers, adders, and memory. It may be possible to vary supply voltage depending on the algorithm being executed to further reduce energy. There may also be opportunities to exploit data relationships to reduce transition activities in an implementation.

If other parts of an MPEG-2 decoder can be designed to reduce significant amounts of energy with such small impact on quality, these techniques can be combined with the IDCT design technique to produce a powerful quality/energy scalable decoder. The idea of trading quality for energy in a manner that exploits the variability of data in terms of its robustness to approximate processing can be used for other modules in an MPEG-2 codec as well as other digital system designs. This paper demonstrates the high potential of such a design methodology.

References

[1] Ludwig, J.T., Nawab, S.H., and Chandrakasan, A.P., "Low Power Digital Filtering using Approximate Processing", IEEE Journal of Solid-State Circuits," pp. 395-400, Mar. 1996.

[2] Goel, M. and Shanbhag, N., "Low-power Channel Coding via Dynamic Reconfiguration," in Proc. Int. Conf. Acoust., Speech, Signal Processing (ICASSP), Phoenix, AZ, pp. 1893-1896, Mar. 1999.

[3] Goodman, J., Chandrakasan, A., and Dancy, A., "Design and Implementation of a Scalable Encryption Processor with Embedded Variable DC/DC Converter", ACM Design Automation Conference (DAC), pp. 855-860, June 1999.

[4] K. Lengwehasatit and A. Ortega,"DCT Computation Based on Variable Complexity Fast Approximations", In Proc. of Int'l Conf. on Image Proc., ICIP'98, Chicago IL, pp. 95-99, Oct. 1998.

[5] P. Duhamel and H. H'Mida, "New 2^n DCT Algorithms suitable for VLSI Implementation," in Proc. Int. Conf. Acoust., Speech, Signal Processing, Dallas, TX, pp. 1805-1808, Apr. 1987.

[6] Z. Wang, "Fast Algorithms for the Discrete W Transform and for the Discrete Fourier Transform", IEEE Trans. on Acoust., Speech, and Signal Processing, vol. ASSP-32, No. 4, pp. 803-816, Aug. 1984.

[7] MPEG Software Simulation Group, "MPEG-2 Video Codec: Test Model 5," http://www.mpeg.org/MPEG/MSSG/, 1996.

[8] Y.Arai, T.Agui, M.Nakajima, "A Fast DCT-SQ Scheme for Images," Trans. of the IEICE, vol. E-71, no. 11, pp. 1095-1097, Nov. 1988.

[9] A. Ligtenberg and M. Vetterli, "A Discrete Fourier/Cosine Transform Chip," IEEE J. on Sel. Areas in Comm., vol. SAC-4, pp.49-61, Jan. 1986.

[10] F. Catthoor et al, "Custom Memory Management Methodology – Exploration of Memory Organization for Embedded Multimedia System Design," Norwell, MA: Kluwer, 1998.

[11] D. Lanneer, M. Cornero, G. Goosesens, H. De Man, "Data Routing: A Paradigm for Efficient Data-path Synthesis and Code Generation," 7th ACM/IEEE International Symposium on High-Level Synthesis, pp. 17-22, May 1994.