DESIGN AND IMPLEMENTATION OF A REAL TIME HIGH QUALITY DV DIGITAL VIDEO SOFTWARE ENCODER

Manish Arora, Shashi Kant and Korada Ramkishor

Emuzed India Pvt. Limited, No. 2 & 3, Jivan Bhima Nagar Main Road, HAL III Stage, Bangalore, India-560075 Email arora, shashi, ramki@emuzed.com

Abstract: *This paper addresses issues in the design and implementation of real time high quality DV video encoder in software. Such software assumes importance in PC based capture-encode-edit-transcode-store applications, which are gaining immense popularity because of the cost advantages with software only systems, possible now due to rapid increase in native signal processing power of a PC. A method to distribute bits within the videosegment to yield the best possible quality at the fixed bit rate is suggested. A novel solution to the problem of macroblock quantization number selection with minimal quantization-variable length encoding cycles is also proposed, which is the major efficiency bottleneck in software DV encoding. Our software encodes faster than real time PAL and NTSC video on a Pentium-4 2.0 GHz PC at high quality.*

Keywords: *DCT, quantization number, class number, video segment, spatial activity.*

1. INTRODUCTION

DV was announced in 1993 [1] aimed at digital home-use recording of high-quality video signals. Subsequently the scheme has been standardized [2] and a number of DV based camcorders have been appearing in the market since 1995. The development of the DV standard is done to satisfy a number of constraints such as easy editing at the picture level, robustness to multiple compression-decompression cycles, easy forward and backward search and playback, high picture quality, etc. DV has been very successful in providing an easy and convenient medium in the above respects, though at the disadvantage in terms of compression when compared with MPEG. In a typical application, DV is used at the acquisition-editing stage and MPEG based formats are used at the storage-distribution stage. The acquisition of DV video is done via the DV camcorders and editing is done on dedicated hardware for professional use.

Today home use consumer electronics designers are moving towards camcorders with PC based software solutions. DV encoded video from the camcorder is transferred to a PC for editing, transcoding and subsequent storage or distribution. This has been possible due to the native signal processing power of the PC, which is expected, to further increase manifold in the coming years. Current systems can easily handle special effects and transcoding [3] to MPEG in software. The proposed algorithm handles real time encoding of raw video into DV video stream on Pentium-4 2.0 GHz PC. Only a simple audio-video digitization card along with a camera would be required for this application bringing down the cost drastically while providing the flexibility of software encoding. Such a card could also be used to capture content from other sources such as VCR, DVD players, etc. and the PC could be used subsequently for DV encoding, editing, special effects and storage.

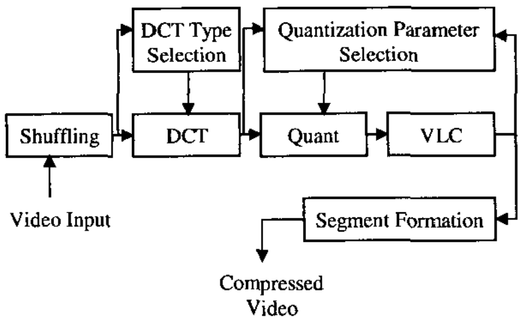

Fig. 1: DV Video Compression Methodology

	Class number				Area number			
	0	1	2	3	0	1	2	3
Quantization number (QNO)	15				1	1	1	1
	14				1	1	1	1
	13				1	1	1	1
	12	15			1	1	1	1
	11	14			1	1	1	1
	10	13		15	1	1	1	1
	9	12	15	14	1	1	1	1
	8	11	14	13	1	1	1	2
	7	10	13	12	1	1	2	2
	6	9	12	11	1	1	2	2
	5	8	11	10	1	2	2	4
	4	7	10	9	1	2	2	4
	3	6	9	8	2	2	4	4
	2	5	8	7	2	2	4	4
	1	4	7	6	2	4	4	8
	0	3	6	5	2	4	4	8
		2	5	4	4	4	8	8
		1	4	3	4	4	8	8
		0	3	2	4	8	8	16
			2	1	4	8	8	16
			1	0	8	8	16	16
			0		8	8	16	16

Fig. 2: Variation of the Quantization Scale

Implementing DV encoding in software imposes a number of challenges due to high input resolution. The DV video encoding standard operates at a fixed bit rate. The bit rate is constant at the video segment level onwards, which is a collection of five macroblocks taken from different portions of the image for content averaging. The first research issue is how to distribute bits within the fixed video segment budget. This will have a significant impact on the quality within a video segment. The second research issue is how to determine the quantization scale that will produce bits after variable length encoding less than the video segment budget but without any wastage. The brute force approach is to try all possible quantization scales, perform variable length encoding on them and take the best-fit quantization scale. But this approach is computationally very expensive in software and is commonly implemented using multiple processing units in hardware [4]. Solutions to these problems along with descriptions on the other portions of the codec are discussed in the rest of the paper.

Section 2 discusses the DV video compression algorithm and identifies the software DV implementation bottlenecks. Section 3 discusses solutions to the research issues on sub-videosegment budget allocation and fast quantization number Selection. Section 4 gives details of the overall encoder implementation and performance. In section 5, results and conclusions are discussed.

2. DV VIDEO COMPRESSION

Figure 1 shows the block diagram of the DV video compression methodology. DV is an advanced intra frame DCT-based compression algorithm. The video input resolution is NTSC (4:1:1 sampled, 720x480 at 30 frames per second) or PAL (4:2:0 sampled, 720x576 at 25 frames per second). Five macroblocks are selected form a frame in pseudo random manner from different locations of the frame using a method specified by the standard. Each macroblock consists of four 8x8 luminance and two 8x8 chrominance blocks. The aim of the selection from different portions of the image is to average the spatial content. Such a selection forms a video segment. This process of video segment formation is called as shuffling. The next step is DCT encoding of the 8x8 blocks in all the macroblocks. DV standard supports 2 DCT encoding methods. Either 8x8 DCT can be applied to the interleaved fields or each of the fields can be DCT transformed separately with a 4x8 transform. The selection method is not specified in the standard. After DCT the coefficients are scaled with a specified method.

Table 1: PSNR with and without sub-videosegment level budgets.

Sequence	PSNR with Bit Budgeting	
	Macroblock Level (Proposed)	Videosegment Level (Brute force)
Formula1	38.185932	35.639852
Display	37.400975	36.331079
Mobile & calendar	30.853673	28.768172
Rowing	41.665969	39.625353
Football	42.260724	40.589910
Suzie	45.363779	43.855863
Stefan	38.397783	35.612923

Table 2: Average number of Quantization-VLE cycles required with estimate only, with estimate and finer adjustments and with the brute force method.

Sequence	Estimate Only	Estimate and Fine adjustments	Brute Force
Formula1	1.3001	1.6333	3.3094
Display	0.9492	1.1610	2.1356
Mobile & calendar	1.2020	1.5649	3.3170
Rowing	1.4403	1.9063	3.8851
Football	1.3820	1.8121	4.0085
Suzie	1.3578	1.8593	3.3693
Stefan	1.3286	1.6397	3.4478

The next step after the DCT is quantization. The quantization process is controlled using two parameters. The first parameter is class number, which controls the quantization scale at a very coarse level. The class number selection is usually done based on the maximum amplitude of the DCT transformed AC coefficients. The class number is specified at the block level and there are 4 possible classes. The second parameter is quantization number (QNO). QNO controls the quantization scales at a finer level once the coarse selection of scale is specified by the class number determination. There are a maximum nine possible quantization numbers that can be varied for each class depending upon the availability of bits keeping in mind the overall budget constraint of the video segment. The QNO can only be varied at the macroblock level. The quantization scale also varies with the zigzag position of the AC coefficients. The coefficients closer to the DC value are quantized with lesser scales and high frequency components are quantized coarsely. The 8x8 DCT block after zigzag scan is divided into 4 portions specified by area numbers. Figure 2 shows the quantization scale variation with class number, QNO and Area Numbers [2].

After the quantization the quantized data is variable length coded. A two-dimensional variable length encoding method is used with the amplitude and run of zeroes before the present coefficient being encoded by variable length codes. The variable length coded, quantized, and zigzag scanned DCT coefficients are then used to form compressed segments. The budget of the compressed segment is fixed at 2560 bits by the standard and there are no restrictions or specifications on how to distribute bits amongst the macroblocks and in the macroblock amongst the blocks. The role of the Quantization Parameter Selection block shown in Figure 1 is to select the class number and QNO such that after the variable length coding the compressed segment budget is not exceeded yet maximally utilized. The simplest way to maximally utilize the bits is to try all the quantization scales one by one starting from the finest quantizer for whole of the videosegment and stop where the bits fit into the fixed budget by reducing the quantization scale of the block taking maximum bits. Each such attempt to find the quantization scales would require quantization and Variable Length Encoding (VLE) of the block data requiring enormous amounts of computation. Though we may find the quantization scales to yield the best fit into the budget but we may not achieve the best quality with this method. In trying to maximize the bits used we also need to make sure that we give the blocks with more spatial activity more number of bits. An integrated approach to distribute bits amongst sub-videosegment units and fast quantization number selection is presented in the next section.

Table 3: PSNR and percentage of bits used with and with only estimate and with estimate and finer adjustments.

Sequence	PSNR		Percentage Bits Used	
	Estimate only	Estimate + Fine	Estimate only	Estimate + Fine
Formula1	38.1040	38.1859	95.76	95.98
Display	37.4198	37.4010	93.27	93.62
Mobile & calendar	30.9476	30.8537	96.47	96.70
Rowing	41.6660	41.6660	97.45	97.71
Football	42.1106	42.2607	96.45	97.09
Suzie	45.2342	45.3638	95.03	94.96
Stefan	38.1081	38.3978	96.49	96.58

Table 4: Profile Information for Encoder without and with SIMD Optimizations.

Module	Percentage Time Spent	
	Before Optimization	After Optimization
Quantization	26.4	16.5
Block Classification	13.6	9.2
DCT	30.0	6.9
Form video segment	6.1	8.6
VLE	10.7	27.0
Frame Reading	3.6	8.7
Miscellaneous	9.6	23.1

3. BIT DISTRIBUTION AND FAST QUANTIZATION NUMBER SELECTION

First of all we determine the spatial activity or measure of content in a DCT transformed block. The more the activity in the block the more bits should be allocated from the video segment budget. The macroblock spatial activity is found as sum of the block activities. By comparing activities and allocating appropriately from the overall video segment budget can determine the macroblock budgets. If the activity measures are appropriate, such a measure would also be highly correlated with the actual number of bits used by the block after quantization and variable length encoding as demonstrated in [5]. We devised such a simple measure of content in a block operating on the DCT transformed data.

The spatial activity is saturated-scaled sum of absolute value of all the AC coefficients. Saturation is done to a maximum value of the AC coefficients since a very high amplitude value will not be much more perceivable than a high value. Scaling is done according to the area numbers with low frequency components being given more importance than high frequency content.

Such content measure is calculated for all the blocks and macroblocks in the video segment and budgets are allocated accordingly. Table 1 shows the PSNR gains by making sub-videosegment budget allocations and then selecting appropriate quantization scales to maximize individual budgets as compared to maximally using the videosegment budget alone by starting at the lowest quantizer level for all blocks and increasing the scale for the block using the maximum bits until the budget is not exceeded. The latter is the most common approach followed in structured hardware implementations.

The next step is to determine the quantization number so that the bits are maximally allocated. We find that our energy measure is strongly correlated with the number of bits as expected and as found for similar energy measures [5]. Figure 3 shows the variation of actual number of bits after variable length encoding for class 3 data with the content measure for different quantization numbers. The relation is almost linear in nature. Statistics are collected over large video data for variation of the bits after variable length encoding with different quantization numbers. Average statistics are collected for different energy levels for each of the class number separately. These tables can now be used to lookup the estimate of quantization numbers depending on the bit budget of the particular macroblocks. Bits used by macroblock after variable length encoding are estimated for each quantization number, by adding individual block estimated bits (from table lookup). The quantization number with closest estimate of bits is taken as initial estimate of quantization number for the macroblock. The macroblock block is next quantized with the estimated quantization number and if the bit budget overflows or underflows further adjustments to the quantization number can be done. Usually a single further modification is sufficient to meet the budget requirements. Table 2 shows the average number of quantizations required with this table lookup method of quantization number selection for different sequences. It also shows the number of quantizations using the brute force approach and after fine adjustments to the estimate for maximizing the budget usage. Here macroblocks having bit budget equal to zero are not considered for quantization.

Table 3 shows the PSNR and percentage of budget utilized with and without finer adjustments to the quantization number estimates. Clearly the difference in PSNR is not much with and without finer adjustments proving that the estimation method predicts close to the actual value after fine adjustments.

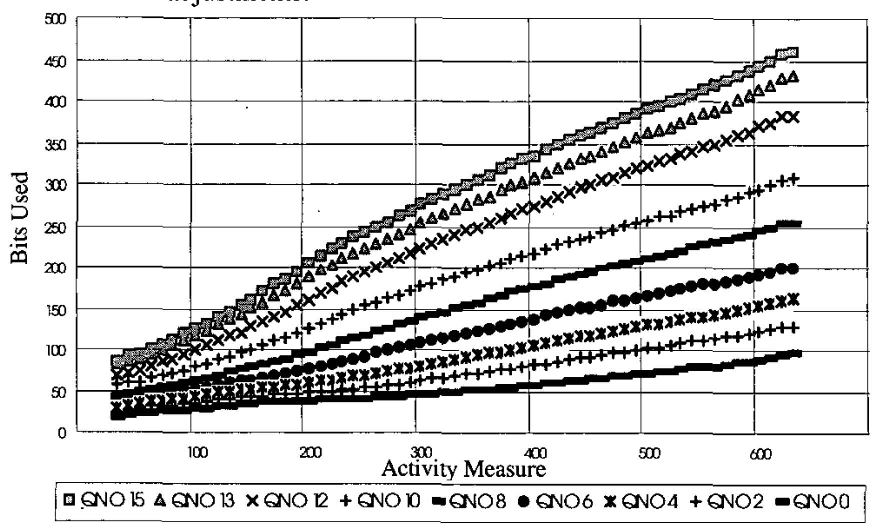

Fig. 3: Variation of Bits with Spatial Activity Measure

4. OVERALL VIDEO ENCODER

A C language reference code for the video encoder is first developed followed by SIMD Pentium-4 specific assembly optimizations. Improvements are made to the finer adjustments of the quantization number selection portion of the algorithm by adaptively increasing and decreasing the present value depending on the amount of overflow and underflow. It was seen that in some places during fine adjustments to the quantization number in trying to maximize the bit budget the overall budget exceeded. Such cases were predicted and previous quantized data was maintained and reverted back to instead of requantizing. All the divisions in the algorithm were converted to fixed-point multiplications to reduce the number of cycles.

The most computation intensive portions of the video encoder are the quantization-variable length encoding, followed by the DCT. 2D-DCT is the ideal candidate for SIMD optimizations when implemented with the row-column DCT approach. Quantization, video segment formation and spatial activity calculation were also implemented in assembly. Table 4 shows the profile information for C code and after SIMD optimizations. Table 5 shows the overall encoded frames per second values achieved for various test sequences on a Pentuim-4 2.0 GHz machine with 256 MB RAM.

Table 5: Frames Per Second for Various Test Sequences.

Sequence	Encoding Frames Per Second	Resolution
Formula1	25.30	PAL
Display	25.26	PAL
Mobile	25.08	PAL
Football	29.86	NTSC
Stefan	29.23	NTSC

5. RESULTS AND CONCLUSIONS

In this paper, we described the design and implementation of DV software encoder. We have highlighted some of the challenges in encoding process and discussed how our encoder implementation addresses these issues. A brute force encoder with efficient convergence towards final quantization number is implemented for comparison. Allocation of sub-videosegment budgets improved the bit distribution over macroblocks in the video segment leading reasonable increase in PSNR. The reduction in number of quantizations and variable length encoding operations using the proposed rate control method is close to 50% when compared to the brute force approach. Optimized SIMD assembly code works real time on Pentium-4 2.0 GHz processor.

REFERENCES

[1]. P. H. N. De, and A. M. A. Rijckaert, "Design considerations of the video compression system of the new DV camcorder standard," *IEEE transactions on consumer electronics, Vol. 34,* No. 4, pp. 1160-1179, November 1997.

[2]. "International Standard, Helical-scan digital video cassette recording system using 6,35 mm magnetic tape for consumer use (525-60, 625-50, 1125-60 and 1250-50 systems)-Part 2": SD format for 525-60 and 625-50 systems, IEC 61834-2.

[3]. Wanrong Lin, Denish Bushmitch, Raghuraman Mudumbai, and Yao Wang, "Design and implementation of high quality DV50-MPEG software transcoder," *Proceedings of ICCE*, LosAngles, June 2002.

[4]. Jeff. Y. F. Hsieh, and Teresa H. Y. Meng, "Low power DV encoder architecture for digital CMOS camcoder," *IEEE Conference on Acoustics Speech Signal Processing*, March 1999.

[5]. W. Lin, D. Bushmitch, Y. Wang and J. Safar, "Bit number prediction for VLC coded DCT coefficients and its application in DV encoding/transcoding", *IEEE workshop on Multimedia Signal Processing,* Cannes, France, October 2001.