Table 2: Rate-constrained PSNR performance for various schemes

Coding scheme	Complexity	Carphone		Foreman	
		Rate	PSNR	Rate	PSNR
	%	bit/frame	dB	bit/frame	dB
BM	100	4063	32.81	4085	32.20
LOG	125	4051	33.14	4073	32.17
DVE	145	4051	33.41	4073	32.61
EXH	1800	4050	33.50	4075	32.63
DVE+GTA	200	4047	33.47	4070	32.65
EXH+GTA	2500	4044	33.54	4070	32.74

was coded using the conventional method. The complexity of each scheme was roughly estimated and expressed as a percentage of the complexity of the conventional scheme. Results obtained by processing the 'Carphone' video test sequence, with $\lambda = 60$ and a quantisation step size of 20 for DCT coding, are shown in Table 1. The DVE algorithm is ~20% more accurate than the LOG approach. The inaccuracy of the DVE algorithm is < 20%, and this drops to < 10% when the block size is reduced to 4×4.

We now compare the average peak signal-to-noise ratio (PSNR) performance of the schemes for a given average bit rate. The performance results for two test sequences are shown in Table 2. The results show that the DVE algorithm has a performance ~0.5dB better than that of the BM method and 0.35dB better than that of the LOG approach, and < 0.1dB poorer than that of the EXH method. We have also included results arising from the use of a generalised form of Ramchandran's algorithm [5] for DCT coding. The modest gains provided by the generalised algorithm (GTA) suggest that DVE obtains most of the gain available for solving eqn. 1.

20 December 1999
Electronics Letters Online No: 20000302
DOI: 10.1049/el:20000302

Jiandong Shen and Wai-Yip Chan (*Department of Electrical and Computer Engineering, Illinois Institute of Technology, 3301 South Dearborn, Chicago, IL 60616-3793, USA*)

E-mail: chan@ece.iit.edu

References

1 GIROD, B.: 'Rate-constrained motion estimation',SPIE Proc. Visual Communications and Image Processing, 1994, Vol. 2308, pp. 1026–1034
2 ITU-T Recommendation H.263, 'Video coding for low bit rate communication'. Draft, November 1995
3 HOANG, D.T., LONG, P.M., and VITTER, J.S.: 'Efficient cost measures for motion compensation at low bit rates'. Proc. Data Compression Conf., 1996, pp. 103–111
4 Telenor Research, 'TMN5 (H.263) encoder/decoder version 2.0'. ftp://bonde.nta.no/pub/tmn, June 1996
5 RAMCHANDRAN, K., and VETTERLI, M.: 'Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility', *IEEE Trans. Image Process.*, 1994, **3**, (5), pp. 700–704

General method for eliminating redundant computations in video coding

L.A. Sousa

A new simple and efficient method for avoiding useless computations in the video coding process is proposed. Experimental results show the practical interest of the method for reducing the computation in software coders and the power consumption in hardware coders.

Introduction: In video coding, video sequences are compressed by exploiting both spatial redundancies (using transformations) and temporal redundancies (using predictive coding together with motion compensation). In recent years, several algorithms have been proposed to reduce the computation required for video coding: fast algorithms for computing the discrete cosine transformation (DCT) and simpler alternative algorithms for the full search (FS), such as the two-dimensional logarithmic search (2D-LS) [1] and the one-at-a-time search (OTS) [2].

This Letter proposes a new general method to reduce the computation required for video coding. This method exploits the distortion value computed during the motion estimation phase to detect blocks for which it is useless to go through the subsequent coding/decoding phases. It can be applied independently of the algorithms used for computing both the DCT and motion estimation by block matching.

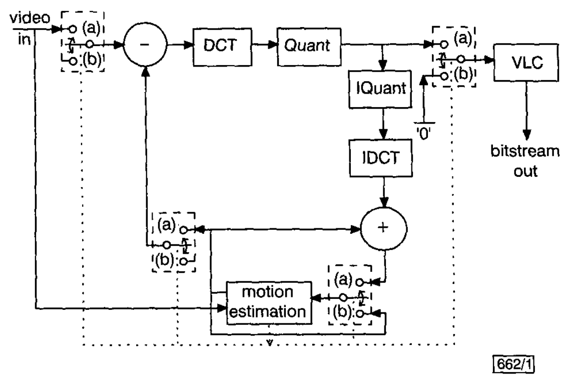

Fig. 1 *Diagram of video coding process shows traditional scheme when switches are in position (a)*

Two-way switches are used for proposed method

Video coding process: Fig. 1 depicts a diagram for the H.263 coding process or one possible implementation for MPEG coding [3], when switches are fixed in position (a). DCT and Quant perform the transformation and the quantisation of the resulting coefficients, respectively, VLC applies variable length coding, while IQuant and IDCT represent the inverse operations. Pictures are usually organised in macroblocks of 16×16 pixels, formed by four 8×8 luminance blocks ($B = 0, ..., 3$), one 8×8 Cr block ($B = 4$) and one 8×8 Cb block ($B = 5$), with chrominance sub-sampled by a factor of two in both the horizontal and vertical directions. In inter-frame (INTER) coding mode, a motion vector (v_x, v_y) is usually computed by applying a block-matching procedure to all four luminance blocks of a macroblock, i.e. by searching the candidate macroblock over a search area (SA) of the reference picture (f_{t-1}) that 'best' matches the macroblock in the current picture (f_t). The best match is usually established by computing the sum of absolute differences (SAD) (eqn. 1). Six difference blocks are then calculated and put through the DCT process (eqn. 2).

$$SAD(i,j) = \sum_{B=0}^{3} SAD^B(i,j) \quad SAD(v_x, v_y) \leq SAD(i,j) \quad \forall (i,j) \in SA$$

$$SAD^B(i,j) = \sum_{x=0}^{7}\sum_{y=0}^{7} |f_t^B(x,y) - f_{t-1}^B(x+i, y+j)| \tag{1}$$

$$F^B(u,v) = \frac{K(u)K(v)}{4} \sum_{x=0}^{7}\sum_{y=0}^{7} [f_t^B(x,y) - f_{t-1}^B(x+v_x, y+v_y)] C(x,u)C(y,v)$$

$$K(\phi) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } \phi = 0 \\ 0 & \text{for } \phi \neq 0 \end{cases}$$

$$C(z,\psi) = \cos\left(\frac{(2z+1)\psi\pi}{16}\right) \qquad 0 \leq B \leq 5 \tag{2}$$

Eliminating redundant computations: In INTER mode, the coding of a given macroblock begins by computing the SAD values and establishing the motion vector (v_x, v_y). At this point, the DCT coefficients for the six difference blocks are quantised with a step size of $\Delta = 2 \times L$ (where Δ is even) and a central dead-zone around zero. This allows us to propose the following theorem.

Theorem 1: Given the $SAD^B(v_x, v_y)$ for a luminance block B (eqn. 1), the DCT coefficients of the difference block are all quantised with the level (L) zero if

$$SAD^B(v_x, v_y) < 8 \times L \times \cos^{-2}\left(\frac{\pi}{16}\right) \qquad (3)$$

Proof of theorem 1: All $F^B(u, v)$ coefficients are quantised with the level zero iff:

$$\left\lfloor \frac{|F^B(u,v)|}{\Delta} \right\rfloor < 1 \Leftrightarrow |F^B(u,v)| < 2L \qquad 0 \le u, v \le 7 \qquad (4)$$

Eqn. 5 can be easily derived from eqn. 2 as follows:

$$|F^B(u,v)| \le \frac{K(u)K(v)}{4} \sum_{x=0}^{7}\sum_{y=0}^{7} |f_t^B(x,y) - f_{t-1}^B(x+v_x, y+v_y)| C(x,u)C(y,v) \quad (5)$$

A maximum for $|F^B(u, v)|$ can be found by noting that:

$$|F^B(u,v)|_{\{u=0,v=0\}} \le \frac{1}{8} SAD^B(v_x, v_y) \qquad (6)$$

$$|F^B(u,v)|_{\{(u=0,v\neq 0)\vee(u\neq 0,v=0)\}} \le \frac{1}{4\sqrt{2}} \cos\left(\frac{\pi}{16}\right) SAD^B(v_x, v_y) \quad (7)$$

$$|F^B(u,v)|_{\{(u\neq 0,v\neq 0)\}} \le \frac{1}{4} \cos^2\left(\frac{\pi}{16}\right) SAD^B(v_x, v_y) \quad (8)$$

By substituting into eqn. 4 the bound for $|F^B(u, v)|$, established in eqn. 8, we derive the sufficient condition for all DCT coefficients quantised with the level zero:

$$\frac{1}{4} \cos^2\left(\frac{\pi}{16}\right) SAD^B(v_x, v_y) < \Delta \qquad (9)$$

and the proof of theorem 1 is completed. QED

The two-way switches in Fig. 1 are used to apply theorem 1 to video coding. For each macroblock, the motion estimation unit receives the value of the quantisation step, computes $SAD(v_x, v_y)$ and stores $SAD^B(v_x, v_y)$, and checks the condition stated in eqn. 3 for each luminance block ($0 \le B \le 3$). If the condition is true, all switches are placed in position (b) and a block of zeros is passed to the VLC unit; otherwise, the switches are placed in position (a). With the switches in position (b), it is useless to perform the subtraction, the DCT, the Quant, the IDCT, the IQuant and the addition. To complete the motion compensation process, the motion estimation unit only has to move the block according to the motion vector. Hardware video coders can also take advantage of the simplified processing diagram to reduce the power consumption, which is an important issue in today's systems. The method

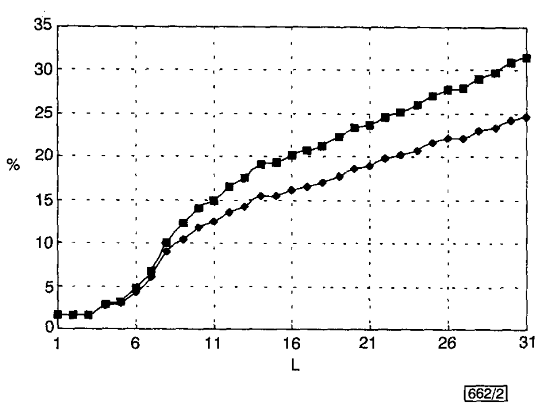

Fig. 2 *Average percentage of blocks not processed for set of benchmark sequences*

Full search algorithm SA = –8/+7
◆ B = 0, ..., 3
■ B = 0, ..., 3, 4, 5

Table 1: Percentage of blocks not processed for Carphone sequence

Search algorithm	Search area	Quantisation level (L)				
		2	4	6	16	30
2D-LS	–8/+7	0.0	0.3	3.1	8.8	17.3
OTS	–8/+7	0.0	1.3	2.8	8.9	17.6
FS	–8/+7	0.0	1.4	4.0	9.8	17.8
	–16/+15	0.0	1.4	4.2	10.2	18.4

Table 2: Percentage of blocks not processed for Claire sequence

Search algorithm	Search area	Quantisation level (L)				
		2	4	6	16	30
2D-LS	–8/+7	5.9	8.2	23.0	35.0	40.1
OTS	–8/+7	5.9	8.6	22.8	35.0	39.6
FS	–8/+7	5.9	8.7	23.9	35.3	40.2
	–16/+15	5.9	8.7	25.0	35.4	40.3

was applied to QCIF format (144 × 176 pixels) benchmark video sequences on an H.263 software coder [4], and each macroblock was coded in INTRA mode once every 40 times. Tables 1 and 2 present the percentage of the total number of blocks (6 × 99) not processed for two video sequences. This percentage grows when the quantisation step increases, but it is nearly constant for the different search algorithms and sizes of the search area. For the video sequence Carphone, which represents a complex scene with a lot of movement, the number of blocks not processed only has a significant value for $L > 4$ and assumes a maximum value of ~17%. For the video sequence Claire, which is a scene with less movement, ~6% (L = 2) to 40% (L = 30) of the blocks are not processed. Fig. 2 depicts the average results obtained for a set of six benchmark video sequences: Carphone, Akiyo, News, Suzie, Claire and Trevor. Approximately 10% of the blocks are not processed for a typical quantisation level of 8, and this number grows continuously until it reaches ~25% for the maximum quantisation level. Although the proposed method cannot be directly applied in chrominance space, we can observe what happens to chrominance difference blocks (B = 5,6) of a macroblock when the condition in theorem 1 is satisfied for all its luminance difference blocks ($0 \le B \le 3$). The conclusion is that they are almost always quantised with level zero, with the exception of two sequences that fail for less than 1% the considered macroblocks. From Fig. 2, it can be concluded that the percentage of blocks not processed increases when taking advantage of the method for the chrominance difference blocks, reaching ~32% for the maximum quantisation level.

Conclusion: A simple and successful method for eliminating useless computations in video coding has been proposed. The application of the method to benchmark video sequences shows that it can be used to design more efficient hardware and software video coders.

24 November 1999
Electronics Letters Online No: 20000272
DOI: 10.1049/el:20000272

L.A. Sousa (*Department of Electrical Engineering, Instituto Superior Técnico/INESC, R. Alves Redol, 9-2º, 1000 Lisboa, Portugal*)

References

1 JAIN, J., and JAIN, A.: 'Image displacement measurement and its application in interframe coding', *IEEE Trans. Commun.*, 1981, **COM-29**, pp. 1799–1808

2 SRINIVASAN, R., and RAO, K.: 'Predictive coding based on efficient motion estimation', *IEEE Trans. Commun.*, 1985, **COM-33**, pp. 888–896

3 BHASKARAN, V., and KONSTANTINIDES, K.: 'Image and video compression standards, algorithms and architectures (Kluwer Academic Publishers, Boston, 1995)

4 ISIDRO, J., COELHO, L., FERREIRA, K., and SOUSA, L.: 'On the development of a video CODEC for low bitrate communication in general purpose computers'. Proc. 7th IASTED Int. Conf. Applied Informatics, Austria, February 1999, pp. 285–288