VIDEO ENCODER COMPLEXITY REDUCTION BY ESTIMATING SKIP MODE DISTORTION

Iain Richardson and Yafan Zhao

Image Communication Technology Group
School of Engineering, The Robert Gordon University,
Schoolhill, Aberdeen, AB10 1FR, UK
i.g.richardson / y.zhao-e@rgu.ac.uk

ABSTRACT

DCT-based CODECs such as MPEG-4 Visual and H.263 "skip" many macroblocks during encoding of typical video sequences (i.e. no coded data is transmitted for these macroblocks). This paper describes an algorithm that predicts macroblocks that are likely to be skipped by the encoder. The algorithm estimates the increase in distortion due to not coding (skipping) each macroblock. Macroblocks that can be skipped with little or no increase in distortion are not coded, resulting in substantial computational savings without significantly affecting rate-distortion performance.

1. INTRODUCTION

Software-only video CODECs based on standards such as H263 [1] and MPEG-4 Visual [2] are used in a wide range of applications and offer advantages such as flexibility and ease of upgrading and distribution. For many applications such as mobile videotelephony and PC-based conferencing, processing resources are at a premium and coding performance may be constrained by computational complexity. It is therefore important to develop methods to enable flexible management of computational complexity and coding performance.

Previous work on reducing computational complexity of video CODECs has included many proposals for "fast" motion estimation algorithms [3,4,5,6] and reduced-complexity DCT implementations [7,8,9]. In [10] we describe a variable-complexity algorithm that enables flexible and accurate management of DCT complexity in a software video encoder and a similar approach to motion estimation complexity is presented in [11]. Other approaches have been proposed for variable-complexity DCT computation [12,13] and motion estimation [14,15]. These papers show that variable-complexity algorithms can reduce computational complexity in a video CODEC, usually at the expense of increased distortion.

A video encoder that conforms to MPEG-4 Visual or H.263 processes data in units of a macroblock (16x16 luma samples and corresponding chroma samples). Macroblocks (MBs) are inter-coded (using motion-compensated prediction from one or more previously encoded video frames) or intra-coded (without any prediction from other frames). After encoding, many inter-coded MBs have zero motion vectors (MV) and/or quantized coefficients (QCoeff). A MB with zero MV and all zero quantized coefficients is skipped (i.e. no coded data is transmitted for this MB). A significant number of inter-coded MBs tend to be skipped, especially for low/medium activity video sequences and/or higher quantizer step sizes.

In [16] we present an algorithm that attempts to predict the occurrence of skipped macroblocks prior to encoding. Correct prediction of a skipped macroblock can save significant computational resources since all of the subsequent processing of the macroblock (motion estimation, transform and quantization, entropy encoding) can be avoided. In this paper we describe an improved algorithm for skipped macroblock prediction. For each macroblock, this algorithm estimates the increase in distortion at the decoder if the macroblock were skipped (compared to the distortion if the macroblock were coded). Macroblocks with zero or small estimated increase in distortion are not processed and are marked as "skipped". The proposed algorithm provides a simple, controllable and robust method of managing computational resources, reducing computation whilst minimising distortion due to occasional incorrect skipping of macroblocks.

2. MACROBLOCK DISTORTION AND SKIP PREDICTION

At the decoder, the residual data is added to motion-compensated reference frame samples to produce a decoded MB. The distortion of the decoded MB relative to the original, uncompressed MB data can be

0-7803-8554-3/04/$20.00 ©2004 IEEE.

approximated by a number of energy measures such as Mean Squared Error (MSE), Mean Absolute Error (MAE) or Sum of Absolute Errors (SAE). SAE is typically calculated during motion estimation and so is readily available in a video encoder. SAE for the luma samples of a decoded MB is given by:

$$SAE_{MB} = \sum_{i,j} \left| a_{ij} - b_{ij} \right| \qquad (1)$$

In this equation, a_{ij} is an original luma sample value, b_{ij} is the corresponding decoded luma sample value and i,j range from 0 to 15. SAE increases (approximately) with decoded image distortion. We use SAE_{diff} , the difference in SAE between a skipped MB and a coded MB, as an estimate of the increase in distortion due to skipping a MB:

$$\text{SAE}_{\text{diff}} = \text{SAE}_{\text{skip}} - \text{SAE}_{\text{noskip}} \qquad (2)$$

SAE_{skip} is the sum of absolute errors between the uncoded MB and the luma data in the same position in the reference frame and is an approximate measure of the distortion **if the MB is skipped** (since the decoder copies the luma data from the reference frame when the MB is skipped). $\text{SAE}_{\text{noskip}}$ is the SAE of the decoded MB (compared with the original, uncoded MB) and is an approximate measure of distortion **if the MB is not skipped**. A low value of SAE_{diff} implies that there is little advantage in coding and transmitting the MB; this could be because there has been no significant change from the reference (previously coded) picture and/or because the coded and decoded MB is significantly distorted. A high value of SAE_{diff} implies that there is a significant benefit in coding the MB (compared with skipping the MB).

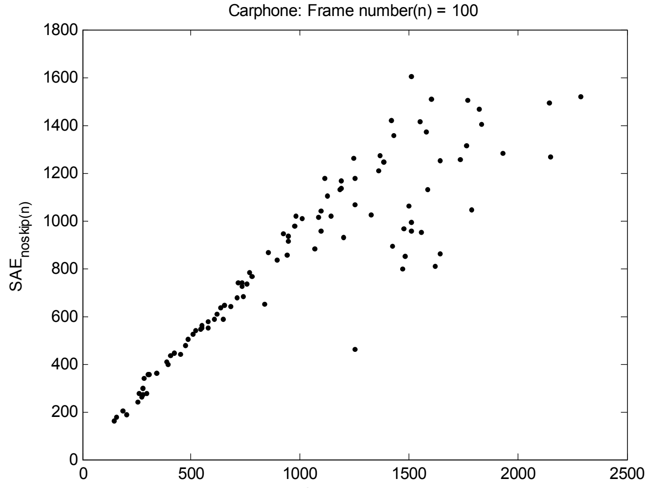

Figure 1 SAEnoskip(i,n) vs. SAEnoskip(i,n-1)

SAE_{skip} is identical to SAE_{00}, typically calculated as the first step of a motion estimation algorithm in the encoder (i.e. the SAE with zero displacement). SAE_{00} (and hence SAE_{skip}) is readily available at an early stage of processing of each MB. $\text{SAE}_{\text{noskip}}$ is the SAE of the decoded MB (compared with the original, uncoded MB) and is not normally calculated during coding or decoding. Furthermore, $\text{SAE}_{\text{noskip}}$ cannot be calculated if the MB is actually skipped. A model for $\text{SAE}_{\text{noskip}}$ is therefore required in order to estimate SAE_{diff} (2). We propose the following model:

$$\text{SAE}_{\text{noskip}}(i,n) \approx \text{SAE}_{\text{noskip}}(i,n-k) \qquad (3)$$

Where i is the current MB number, n is the current frame and $\text{SAE}_{\text{noskip}}(i,n-k)$ is the SAE of the most recent available decoded MB in position i. If the MB in position i in the previous frame was coded (not skipped), then k is 1; if the MB in position i in frame (n-1) was skipped, then a stored $\text{SAE}_{\text{noskip}}$ value from an earlier frame is used.

Figure 1 plots $\text{SAE}_{\text{noskip}}(i,n)$ vs. $\text{SAE}_{\text{noskip}}(i,n-1)$ for frame 100 of the sequence "Carphone" (coded using H.263 TMN10). This Figure demonstrates a clear correlation between the two quantities and so $\text{SAE}_{\text{noskip}}(i,n-k)$ is therefore a good predictor[1] for $\text{SAE}_{\text{noskip}}$. This model requires the encoder to compute $\text{SAE}_{\text{noskip}}$ (a single calculation of equation (1) for each coded MB). The encoder calculates and stores $\text{SAE}_{\text{noskip}}$ for each coded MB, so that older values of $\text{SAE}_{\text{noskip}}$ for position i are replaced when a new MB is coded.

Based on Equation (2) and using the model described above, we propose the following algorithm for selectively skipping (not processing) MBs. In this algorithm, SAE_{diff} is approximated by (SAE_{00}-$\text{SAE}_{\text{noskip}}${estimate}).

MB Skip Model:

```
$\text{SAE}_{\text{noskip}}\{\text{estimate}\} = \text{SAE}_{\text{noskip}}(i,n-k)$
if ($\text{SAE}_{00} - \text{SAE}_{\text{noskip}}\{\text{estimate}\}$) < T
    skip current MB
else
    code current MB
```

T is a threshold that controls the proportion of skipped MBs. A higher value of T results in an increased number of skipped MBs (but also in an increased distortion due to incorrectly skipped MBs).

---

[1] Other predictors for $\text{SAE}_{\text{noskip}}$ (spatial and spatio-temporal median predictors) were evaluated but this predictor was found to be the most effective of those tested.

3. EXPERIMENTAL RESULTS

The standard video sequences “Carphone”, “Mother and Daughter” and “Foreman” were encoded with the H.263 TMN10 encoder and with a modified version of this encoder that incorporates the MB Skip Model described above.

Figure 2 demonstrates the trade-off between computational saving (due not processing MBs) and increased distortion (due to skipping MBs that would normally be coded). The “Carphone” sequence was coded with the MB Skip Algorithm algorithm with a fixed quantizer step size Q (8 or 16) and with increasing values of threshold T. The Figure plots the computational complexity (measured in terms of number of processed macroblocks) against the PSNR drop (i.e. the drop in PSNR compared with the same sequence coded using the TMN10 encoder). As the threshold increases (moving to the left and upwards on the graph), the percentage of MBs skipped by the algorithm increases, causing the computation to drop. At the same time, the number of “false alarms” (skipped MBs that should have been coded) increases and so the PSNR drop increases (compared with the TMN10 encoder). Figure 2 compares the performance of the new “MB Skip Model” algorithm with our previous algorithm described in [16]. The graph demonstrates that the new model out-performs our previous published algorithm (i.e. it achieves a lower computational complexity for the same PSNR drop).

The performance of the MB Skip Model is plotted in Figure 3 for each frame of “Carphone” with the control parameter T chosen in order to skip 45% of macroblocks (i.e. only 55% of macroblocks are processed). The Figure shows the PSNR of each decoded frame, compared with the unmodified TMN10 encoder. The decoded PSNR drops by up to approximately 0.6dB compared with TMN10.

Figure 4 shows the effect of the MB Skip Model on rate-distortion performance, with the threshold chosen so that only 55% of macroblocks are processed. The Model introduces a small rate-distortion penalty at higher bitrates and at low bitrates the rate-distortion performance is actually slightly **better** than the TMN10 encoder. This is because an increase in skipped MBs results in a smaller number of coded bits and, with careful selection of skipped MBs (i.e. choosing to skip MBs that do not significantly improve decoded picture quality), the reduction in bitrate tends to “balance out” the increased distortion.

Similar performance results were obtained with the standard test sequences “Foreman” and “Mother and Daughter”. In each case, a different choice of T was required to obtain a similar proportion of skipped MBs.

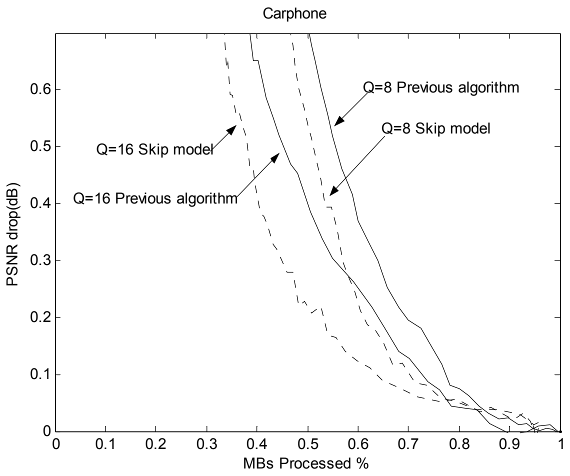

Figure 2 Macroblocks processed vs. PSNR drop

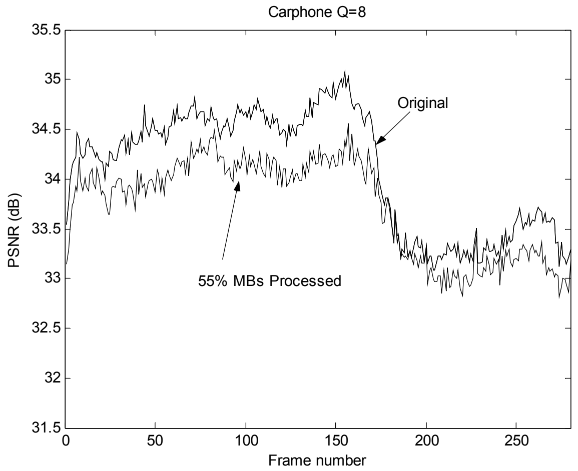

Figure 3 PSNR vs. frame number, Carphone

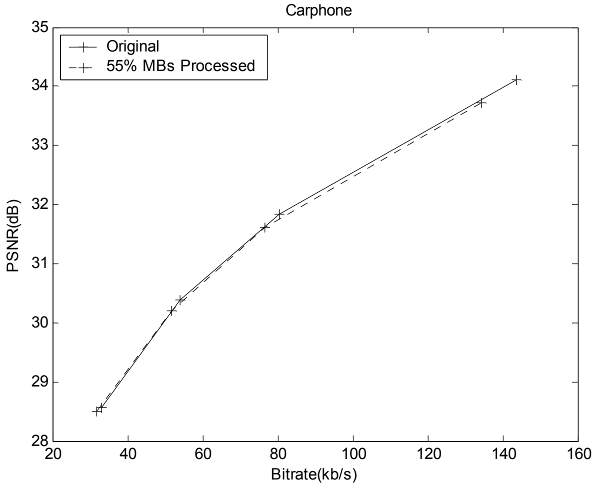

Figure 4 Rate-distortion performance

Computational complexity

The purpose of our proposed algorithm is to reduce the computational complexity during encoding and so it is important to identify any additional computational costs due to the classification algorithm. The MB Skip Model requires the calculation of SAE_{noskip} for each coded (not skipped) MB. This is not calculated in the TMN10 encoder and so is an extra computational cost. The increase in computation required to calculate SAE_{noskip} was determined by comparing the total coding time of a 200–frame sequence using (i) the TMN10 encoder and (ii) the TMN10 encoder with code added to calculate SAE_{noskip}. Calculating this quantity increased coding time by 0.4 – 0.6%, a negligible computational cost compared with the saving due to not processing a significant number of MBs.

4. CONCLUSIONS

This paper describes a method of computational complexity reduction in which a video encoder selectively codes MBs based on an estimate of the increased distortion due to MB skipping. The aim of this method is to save computation through not processing MBs that (a) are likely to be skipped and/or (b) do not significantly improve the decoded image quality. The encoder estimates the increase in distortion (SAE_{diff}) due to skipping (rather than coding) each MB and uses this estimate to determine whether the MB should be processed. The MB Skip Model described here can produce a significant reduction in MB processing with only a modest rate distortion penalty compared with TMN10 (and even a small improvement at low bitrates). The model requires additional computation of a parameter SAE_{noskip} but this computational cost is negligible compared with the reduction in MB processing.

The algorithm presented here provides an effective and controllable method of reducing computation in a video encoder. Further work is required investigate other prediction methods for the parameter SAE_{noskip} , optimal methods of choosing the threshold T and real-time implementation of the algorithm. The work described in this paper is the subject of UK patent application 0229354.6.

5. REFERENCES

1 ITU-T Recommendation H.263, "Video coding for low bit rate communication", 1998

2 ISO/IEC 14496-2, "Information technology – Generic coding of audio-visual objects – Part 2: Visual" (MPEG4), 1999

3 J. Jain and A. Jain, “Displacement measurement and its application in interframe image coding”, IEEE Trans. on Communications, vol. COM-29, Dec. 1981

4 K. I. T. Koga, A. Hirano, Y. Iijima, and T. Ishiguro., “Motion-compensated interframe coding for video conferencing”, in Proc. NTC 81, vol. 28, (New Orleans), Dec. 1992

5 M. Gallant, G. Côté and F. Kossentini, “An efficient computation-constrained block-based motion estimation algorithm for low bitrate video coding”, IEEE Trans. Image Processing 8(12), December 1999

6 J-F Yang, S-C Chang and C-Y Chen, “Computation Reduction for Motion Search in Low Rate Video Coders”, IEEE Trans. Circuits and Systems for Video Technology, Vol. 12 No. 10, October 2002

7 A. Skodras, "Fast Discrete Cosine Transform Pruning", IEEE Trans. Sig. Proc., July 1994

8 A. Hossen and U. Heute, "Fast Approximate DCT", Proc. ICASSP97, Munich, 1997

9 M-T Sun and I-M Pao, “Statistical Computation of Discrete Cosine Transform in Video Encoders”, J. Visual Communications and Image Representation, June 1998

10 Iain E G Richardson and Yafan Zhao, “Adaptive management of Video Encoder complexity”, Journal of Real-time Imaging, Vol. 8, No. 4, Aug 2002, pp. 291-301

11 Yafan Zhao and Iain E G Richardson, "Computational Complexity Management of Motion Estimation in Video Encoders", Proc. IEEE DCC2002

12 Lengwehasatit, K. and A. Ortega, *DCT Computation Based on Variable Complexity Fast Approximations.* Proc. ICIP98, Oct. 1998.

13 R.L. de Queiroz, *Variable complexity DCT approximations driven by an HVQ-based analyzer.* IEEE Trans. on Circuits and System for video technology, Nov. 2002. **12**(11): p. 1021-1024.

14 K. Lengwehasatit and A. Ortega, *A novel computationally scalable algorithm for motion estimation.* Proc. VCIP1998, Jan. 1998

15 R. Braspenning, G.d. Haan, and C. Hentschel, *Complexity scalable motion estimation.* Proc. VCIP 2002, Jan.2002.

16 Yafan Zhao and Iain E G Richardson, “Macroblock skip-mode prediction for Complexity control of Video Encoders”, Proceedings of IEE Visual Information Engineering, July 2003.