# COMPUTATIONAL COMPLEXITY REDUCTION OF INTRA-FRAME PREDICTION IN MPEG-2/H.264 VIDEO TRANSCODERS

Gerardo Fernández-Escribano, Pedro Cuenca, Luis Orozco-Barbosa and Antonio Garrido

Instituto de Investigación en Informática
Universidad de Castilla-La Mancha, Campus Universitario, 02071 Albacete, SPAIN
[gerardo,pcuenca,lorozco,antonio]@info-ab.uclm.es

## ABSTRACT

*MPEG-2 is the most widely digital video-encoding standard in use nowadays. It is being widely used in the development and deployment of digital TV services, DVD and video-on-demand services. However, recent developments have given birth to the H.264/AVC, offering better bandwidth to video quality ratios than MPEG2. It is expected that the H.264/AVC will take over the digital video market, replacing the use of MPEG-2 in most digital video applications. The complete migration to the new video-coding algorithm will take several years given the wide scale use of MPEG-2 in the market place today. This creates an important need for transcoding technologies for converting the large volume of existent video material from the MPEG-2 into the H.264 format and vice versa. However, given the significant differences between the MPEG-2 and the H.264 encoding algorithms, the transcoding process of such systems is much more complex to other heterogeneous video transcoding processes. In this paper, we introduce and evaluate a novel intra-frame prediction algorithm to be used as part of a high-efficient MPEG-2 to H.264 transcoder. Our evaluation results show that the proposed algorithm considerable reduces the complexity involved in the intra-frame prediction: a key operation in the transcoding process.*

## 1. INTRODUCTION

Nowadays, the MPEG-2 video coding format [1] is being widely used in a number of applications from digital TV systems to video-on-demand services. The use of MPEG-2 technology represents billions of dollars of investment in the MPEG-2 infrastructure already or currently being deployed.

During the last few years, technological developments, such as novel video coding algorithms, lower memory costs, and faster processors, are facilitating the design and development of highly efficient video encoding standards. Among the recent works in this area, the H.264 video encoding standard, also known as MPEG-4 AVC occupies a central place [2]. The H.264 standard, jointly developed by the ITU-T and the MPEG committees, is highly efficient offering perceptually equivalent quality video at about 1/3 to 1/2 of the bitrates offered by the MPEG-2 format. These significant bandwidth savings open the market to new products and services, including HDTV services at lower bitrates. Furthermore, given the relatively early stage of video services in mobile phones, mobile phones will be one of the first market segments to adopt H.264 video. However, these gains come with a significant increase in encoding and decoding complexity [3].

While the H.264 video standard is expected to replace MPEG-2 video over the next several years, a significant amount of research needs to be done in developing efficient encoding and transcoding technologies. The transcoding of MPEG-2 video to H.264 format is particularly interesting given the wide availability and use of MPEG-2 video nowadays. Furthermore, there is a clear industry interest in technologies facilitating the migration from MPEG-2 to H.264. The coexistence of these technologies until the complete adoption of H.264 creates a need for technologies to transcode from the MPEG-2 into the H.264 format and vice versa. However, given the significant differences between the MPEG-2 and the H.264 coding algorithms, transcoding is a much more complex task compared to the task involved in other heterogeneous video transcoding architectures [4-8].

The H.264 employs a hybrid coding approach similar to that of MPEG-2 but differs significantly from MPEG-2 in terms of the actual coding tools used. The main differences are: 1) use of an integer transform with energy compaction properties; 2) an in-loop deblocking filter to reduce block artifacts; 3) multi-frame references for inter-frame prediction; and 4) intra-frame prediction. The H.264 standard introduces several other new coding tools aiming to improve the coding efficiency.

In this paper, we focus our attention on the intra-frame prediction: one of the most stringent tasks involved in the encoding process. A complete overview of the H.264 can be found in [9]. The rest of the paper is organized as follows. Section 2 provides a brief overview of the intra-frame prediction process used by the H.264 encoding standard. In Section 3, we introduce a fast intra-frame prediction algorithm suitable for the transcoding of MPEG-2 into H.264. In Section 4, we carry out a performance evaluation of the proposed algorithm in terms of its computational complexity and rate-distortion results. Finally, Section 5 concludes the paper.

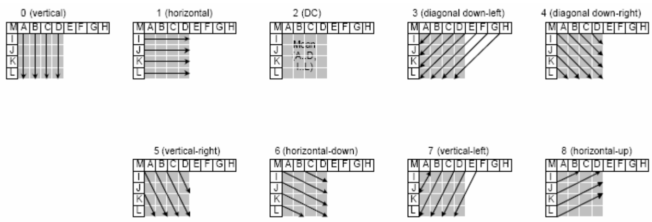
This work was supported by the Ministry of Science and Technology of Spain under CICYT project TIC2003-08154-C06-02, the Council of Science and Technology of Castilla-La Mancha under project PBC-03-001 and FEDER.

*0-7803-9332-5/05/$20.00 ©2005 IEEE*

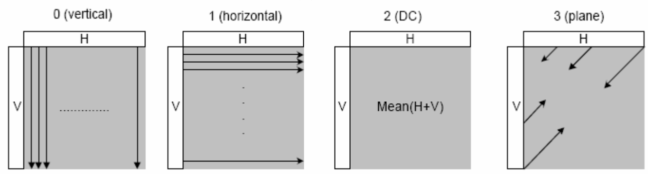
## 2. INTRA-FRAME PREDICTION IN H.264

H.264 incorporates into its coding process, an intra-picture prediction (defined within the pixel domain) whose main aim is to improve the compression efficiency of the intra-coded pictures and intra-MBs. Intra prediction can result in significant savings when the motion present in the video sequence is minimal and the spatial correlations are significant. Throughout the paper, we will illustrate the principle of operation of the intra-frame prediction modes as applied to the luminance blocks. It is understood that a similar procedure has to be applied to the chrominance blocks.

While macro blocks (MB) of 16x16 pixels are still used, predicting a MB from the previously encoded MBs in the same picture is new in H.264. An MB may make use of 4x4 and 16x16 block prediction modes, referred to as Intra_4x4 and Intra_16x16, respectively. There are nine 4x4 possible block prediction directions and four 16x16 block prediction directions. Figure 1 depicts the nine and four prediction directions for the 4x4 and 16x16 prediction modes, respectively. These intra prediction modes include a directional prediction greatly improving the prediction in the presence of directional structures. With the intra-frame prediction, the I-pictures can be encoded more efficiently than in MPEG-2, which does not support intra-frame prediction.




(a)




(b)

Figure 1. Prediction Modes. (a) 4x4 b) 16x16.

For each MB, one prediction mode and one prediction direction is kept. The H.264 encoder selects the best combination mode/direction by using the *Sum of Absolute Errors* (SAE). This implies, that for each existing direction of each mode, the predictor within the pixel-domain is created from the boundary pixels of the current partition and the SAE costs are evaluated. The best combination of mode/direction is determined corresponding to the one exhibiting the minimum SAE cost. The residual is encoded using a 4x4 integer based transform. In the next section, we present a fast intra-frame prediction algorithm suitable for transcoding video material from the MPEG-2 into the H.264 format. We achieve very high computational savings by accelerating the estimation process of intra-frame prediction of H.264 using the DC coefficient of the MPEG-2 DCT 8x8 blocks.

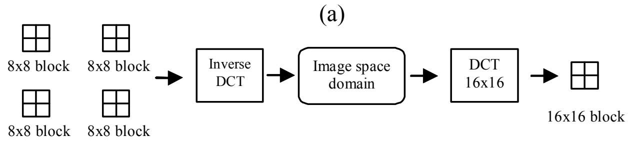
## 3. SPEEDING-UP THE INTRA-FRAME PREDICTION

Our approach simplifies the intra-frame prediction by making use of the DC coefficients available from the decoding process of the MPEG-2. However, due to the presence of two different sizes of blocks used by the H.264, namely 4x4 and 16x16, and that the MPEG-2 standards use blocks of 8x8, the evaluation of the prediction mode involves and intermediate scaling process. In the following, we describe one by one the main steps of our algorithm.

### Step 1:

In an MPEG-2/H.264 video transcoder, once having decoded the MPEG-2 video, besides the uncompressed video, the DC coefficient of the 8x8 blocks is readily available to the H.264 video encoder. Since the MPEG-2 makes use of only 8x8 blocks, we need to devise a mechanism allowing us to properly compute the DC coefficients of the 4x4 and 16x16 blocks. Figures 1a and 1b depict the procedure for computing the DC coefficients of the four 4x4 blocks and the one associated to the 16x16 block.


(a)




(b)

Figure 2. Straightforward method. (a) 4x4. (b) 16x16.

As seen from Figure 2a, the process to obtain the four 4x4 blocks involves first applying the inverse DCT to each 8x8 block of the decoded MPEG-2 picture. This step regenerates the 8x8 block in the space domain. Then, the DCT has to be applied to each 4x4 block in order to obtain the DC coefficients of each one of them. This process has to be applied to each and every 8x8 block. In order to speed-up this process, we propose using the following resolution conversion method introduced in [10]:

$$F'_I = A\, F_I A^T \qquad (1)$$

$$A = \begin{bmatrix} 0.71 & 0.64 & 0 & 0.22 & 0 & 0.15 & 0 & -0.13 \\ 0 & 0.29 & 0.71 & 0.56 & 0 & -0.25 & 0 & -0.20 \\ 0 & -0.05 & 0 & 0.36 & 0.71 & 0.54 & 0 & -0.27 \\ 0 & 0.02 & 0 & -0.07 & 0 & 0.35 & 0.71 & 0.61 \\ 0.71 & -0.64 & 0 & 0.22 & 0 & -0.15 & 0 & 0.13 \\ 0 & 0.29 & -0.71 & 0.56 & 0 & -0.25 & 0 & 0.20 \\ 0 & 0.05 & 0 & -0.36 & 0.71 & -0.54 & 0 & 0.27 \\ 0 & 0.02 & 0 & -0.07 & 0 & 0.35 & -0.71 & 0.61 \end{bmatrix}$$

In simple words, this process transforms the NxN DCT coefficients ($F_I$) into four N/2xN/2 DCT coefficients ($F'_I$) by using the kernel matrix $A$ (see Figure 3a.)

This product of matrices is significantly less complexity than the procedure depicted in Figure 2a: the four DC coefficients are simply obtained from the $F'_I$ matrix. Furthermore, the $A$ matrix contains many zero-valued elements.

Regarding the computation of the DC coefficient of the 16x16 block, this one can be obtained as follows:

$$DC_{16} = \frac{DC_8^1 + DC_8^2 + DC_8^3 + DC_8^4}{2} \qquad (2)$$

this is to say, by adding the four DC coefficients of the four corresponding 8x8 blocks and then dividing the result by two. Equation 2 is simply derived from the fact that the DC coefficient of an NxN block is nothing else but the mean value of all the pixels within the block. This conversion procedure is depicted in Figure 3b.

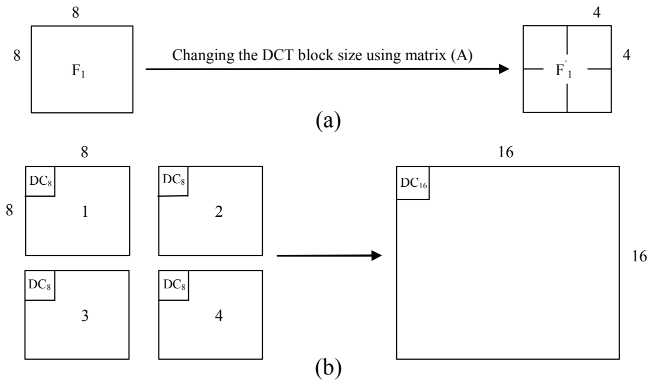



Figure 3. Resolution conversion method
(a) 8x8 to 4x4, (b) 8x8 to 16x16

### Step 2:

The computation of the DC coefficient of the intra luma predictors of the H.264 standard is a straightforward procedure. Let's take the example of computing the *Vertical Predictor* involved in the 4x4 intra luma prediction. The predictor is created by copying the values of the upper border pixels into all the entries within the same column (see Figure 4). According to the DCT, the DC coefficient of the predictor is given by:

DC = a + b + c+ d.

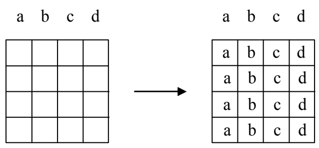



Figure 4. Example of creation of a 4x4 predictor

In this simple form, we are able to compute the DC coefficients of the *Vertical Prediction*. Similarly, this process can be applied for obtaining all the other predictors.

### Step 3:

The third and last step of our proposed algorithm consists in keeping the prediction mode and its corresponding direction whose DC coefficient exhibits the lowest absolute difference with respect to the DC coefficient of the original block. As a further feature allowing us to speed up this process, we may only consider the use of the prediction directions 0, 1 and 2, for both Intra_4x4 and Intra_16x16. We base this choice by the fact that having studied a large number of images available in the database (more than 120.000 samples) reported in [11], these three modes are used in more than 50% of the times.

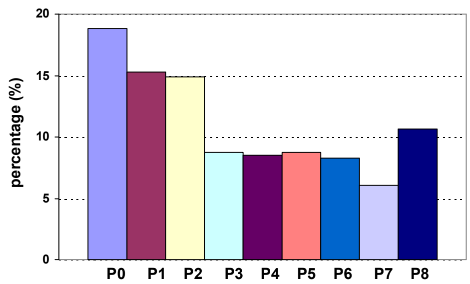



Figure 5. Histogram of H.264 intra predictor directions.

As we will show in the following section, the proposed algorithm will significantly reduce the number of operations involved in the calculation of the intra predictors when compared to the more classical method of computing the differences on a pixel per pixel basis.

## 4. PERFORMANCE EVALUATION

In order to evaluate our *Fast Intra-Frame Mode Decision Algorithm*, we have implemented the proposed approach based on the H.264 reference software [12] (version 9.2). The metrics we have been interested on are the computational cost and rate distortion function. Throughout our experiments, we have used various video sequences exhibiting different spatial characteristics and different size formats (CCIR, CIF and QCIF). We use Q factors from QP=0 to QP=50 (corresponding to the full H.264 QP range). Every frame of each sequence was encoded as I-frame in order to obtain results for intra-frame prediction only.

Table I. Computational Cost.

| | Full Estimation | 4x4 Proposed | 16x16 Proposed |
|---|---|---|---|
| **Additions** | 33326 | 1088 | 39 |
| **Multiplications** | 1075 | 1120 | 1 |
| **Comparisions** | 14342 | 192 | 12 |
| **Divisions** | 5217 | 0 | 1 |
| **Total per MB** | 53960 | 2400 | 53 |
| **QCIF image** | 5342040 | 237600 | 5247 |
| **CIF image** | 21368160 | 950400 | 20988 |
| **CCIR image** | 72846000 | 3240000 | 71550 |

Table I shows the mean number of operations per MB and per image used for the H.264 *full estimation* approach and for the

*Fast Intra-Frame Mode Decision* algorithm proposed using Intra_4x4 and Intra_16x16, showing the high gains on the reduction of computational complexity characterizing our proposed scheme.

Figures 6 to 8 show the RD results of applying the full estimation algorithm and our proposed intra-frame prediction algorithm to six different video sequences. As seen from the figures, the PSNR obtained when applying our algorithm deviates slightly from the results obtained when applying the considerable more complex full estimation procedure. As expected, the difference is less noticeable at lower bit rates: the blocking effect is more noticeable, i.e. the DC coefficient has a heavier weight. Based on the results depicted in Figure 6 to 8, depending on the image quality requeriment, the use of Intra_16x16 prediction mode may prove a viable solution when computational cost may be an issue.

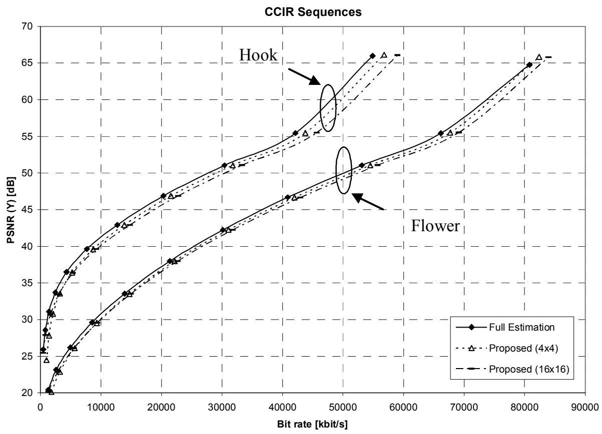



Figure 6. Rate Distortion Results. CCIR sequences.

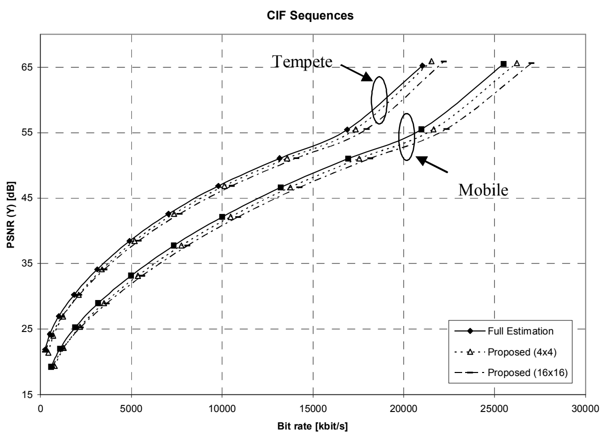



Figure 7. Rate Distortion Results. CIF sequences.

## 5. CONCLUSIONS

In this paper, we have defined a new fast intra-frame prediction algorithm to be used in the implementation of MPEG-2 to H.264 transcoders. Our results show that the proposed algorithm is able to maintain a good picture quality while considerably reducing the number of operations to be performed.

The proposed algorithm can be used as basis for a full low complexity transcoder aplicable in the full QP range. Furthermore, it can be used in the definition of QP based rate control mechanisms. Our future plans include implementing an efficient and dynamic method to select the best prediction mode among the Intra_4x4 and Intra_16x16 prediction modes generated by our algorithm.

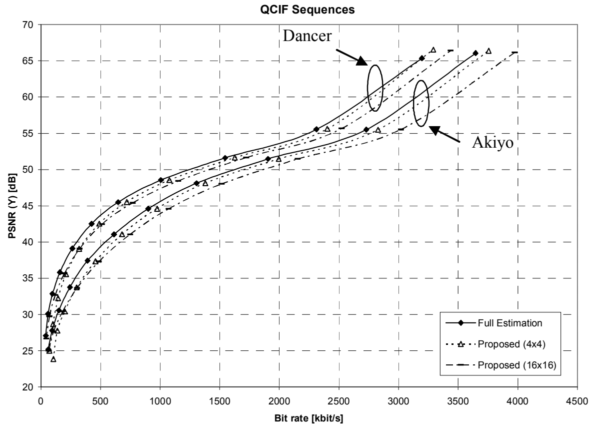



Figure 8. Rate Distortion Results. QCIF sequences.

## 6. REFERENCES


[1] ISO/IEC JTC11/SC29/WG11, "Generic Coding of Moving Pictures and Associated Audio Information: Video", ISO/IEC 13818-2. May 1994.

[2] ITU-T RECOMMENDATION H.264 "Advanced Video Coding for Generic Audiovisual Services". May 2003.

[3] Implementation Studies Group, "Main Results of the AVC Complexity Analysis," MPEG Document N4964, ISO/IEC JTC11/SC29/WG11, July 2002.

[4] N. Bjork and C. Christopoulos, "Transcoder Architectures for Video Coding," *IEEE Trans. Consumer Electronics*, vol. 44, no. 1, pp.88-98, Feb. 1998.

[5] A. Vetro, C. Christopoulos, and H.Sun "Video Transcoding Architectures and Techniques: An Overview". *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp.18-29, March. 2003.

[6] S. Dogan and A. Sadka. "Video Transcoding for Inter-Networks Communications". *Chapter of Compressed Video Communications*. John Wiley & Sons, pp.215-256, March. 2003.

[7] H. Kalva, A. Vetro, and H. Sun, "Performance Optimization of the MPEG-2 to MPEG-4 Video Transcoder," *SPIE Conference on Microtechnologies for the New Millennium, VLSI Circuits and Systems*, May 2003.

[8] J.Bialkowski, A. Kaup and K. Illgner. "Fast Transcoding of Intra Frames between H.263 and H264". *IEEE International Conference on Image Processing*. October 2004.

[9] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra. "Overview of the H.264/AVC Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 7, July 2003.

[10] T. Goto, T. Hanamura, T. Negami and T. Kitamura. "A Study on Resolution Conversión Method using DCT Coefficients". *International Symposium on Information Theory and its Applications*. October 2004.

[11] http://sampl.eng.ohio-state.edu/~sampl/database.htm.

[12] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, *Referente Software to Comitee Draft*. JVT-F100 JM9.2.