AN H.264/AVC INVERSE TRANSFORM ADAPTATION METHOD FOR VIDEO STREAMING APPLICATIONS

Rafael Galvão de Oliveira[*], *Maria Trocan*[†], *Béatrice Pesquet-Popescu*[*]

galvao@telecom-paristech.fr, maria.trocan@isep.fr, beatrice.pesquet@telecom-paristech.fr

ABSTRACT

Bandwidth sharing in video streaming applications becomes problematic with the explosion of live services. In such applications, compression optimization techniques are seen as an efficient way for enforcing high quality video transmission. In this paper, we propose a low-complexity adaptive encoding framework for H.264/AVC, in which the inverse transformation matrix is optimized for each frame, as function of both content and quantization noise. As the transform optimization is done in one encoding pass, it can be successfully used in live encoding setups. The proposed synthesis filter adaptation method shows promising results compared to H.264/AVC, making it suitable for video streaming applications.

Index Terms— Inverse transform, adaptive coding, DCT, filter bank, H.264/AVC.

1. INTRODUCTION

Nowadays, with the expansion of video-based applications, real-time services like live video streaming, video conferencing, telephony or gaming are of great demand. However, as the available bandwidth is limited in such applications, compression optimization techniques represent the best way to achieve high quality video streaming services. Subband decomposition is an efficient technique that can be successfully used in mobile applications, providing the best trade-off between image resolution, framerate and video quality. Originally designed with the filter bank (FB) formalism [1], subband transforms have become the most common tool for video compression due to block-based processing (e.g. DCT) that permits efficient implementations.

In [2] it has been shown that perfect reconstruction (PR) synthesis filter banks do not provide the best performance in terms of the reconstruction quality, when considering the presence of quantization noise. Indeed, the minimization of the quantization error leads to a solution where the synthesis filter bank depends both on the statistics of the quantization noise and of the input signal. In [3], this optimization was done for a 2D signal, firstly decorrelated with a separable 2D-DCT transform and afterwards reconstructed using an adaptive non-separable 2D transform. Using a quantization matrix similar to the one employed by the JPEG [4] standard, it has been shown that it is possible to reconstruct the signal with a substantial signal to noise ratio (SNR) gain as compared with the classical decoding method using a PR-FB. Similarly, in [5] a 2D separable synthesis filter bank was proposed for the optimal reconstruction of JPEG-coded images.

However, in [2] the synthesis FB adaptation is performed as a bitrate constrained optimization, which amounts to jointly optimize the quantizer choice and the synthesis FB design. For a closed-loop hybrid video compression scheme, the optimization of the decoding stage, including the choice of an adaptive inverse transform, is a much more complicated problem. Indeed, the values of the encoded residual coefficients do not only depend on the quantizer but they are also related to the way the residual is obtained, given the prediction modes and motion vectors. In other words, the optimization of the reconstruction stage in a video codec involves a joint optimization of the quantizers, inverse transform, prediction modes and motion vectors [6]. Due to the complexity of the problem, an exhaustive search for the optimal parameters is not tractable and sub-optimal solutions are usually implemented. Firstly, the inverse transform classically used by the decoder corresponds to the synthesis bank filter obtained with the perfect reconstruction property. Next, a widely accepted method for prediction modes and motion vectors selection is to minimize a Lagrangian cost criterion *prior* to performing the residual coding [7]. Both rate and distortion for the actual residual coding stage are approximated in this latter step.

In [8] was introduced an in-loop inverse transform optimization process wherein the synthesis filter bank is adapted for each frame, in a classical rate-distortion optimization scheme. At each iteration, a new inverse transform is computed, by minimizing the mean square reconstruction error (MSE) and, at the end of the iteration loop, a rate-distortion criterion was used to choose the most suitable inverse transform to be used in the reconstruction.

In this paper, we propose to perform the adaptation of the synthesis filter banks outside the closed loop, thus strongly reducing the complexity on the encoder side. This way, by re-

[*] Télécom ParisTech, 46 rue Barrault, 75634 Paris, France.
[†] Institut Supérieur d'Électronique de Paris, 28 rue NDC, 75006 Paris.

moving the necessity of multiple encoding passes in the adaptation process of [8], the new coding scheme can be successfully used on live encoding setups, such as video conferencing or gaming services. Moreover, the resulted stream is completely compatible with the H.264/AVC standard [9], with an optional adapted inverse transform. As the proposed scheme presents interesting gains with respect to H.264/AVC, it could be efficiently used in video streaming applications.

The paper is organized as follows: in the next section, we briefly introduce the synthesis FB formalism and formulate the mean-square error minimization problem. Section 3 describes our inverse transform adaptation algorithm in an H.264/AVC video coding framework. In Section 4, the performance of the proposed method is evaluated, conclusions and perspectives for future works being drawn in Section 5.

2. MSE MINIMIZATION FOR SYNTHESIS FILTER BANKS

Let an input signal x be decomposed into M subbands $\{y_i\}_{0\leq i<M}$ by an M-band analysis filter bank having the impulse response $\{h_i\}_{0\leq i<M}$, followed by decimators of a factor M. Each subband is further quantized: $y_{b,i}(m) = y_i(m) + b_i(m)$, then reconstructed using the synthesis filter bank defined by its $\{g_i\}_{0\leq i<M}$ impulse response. Here $b_i(m)$ denotes the quantization noise of the i^{th} subband and m represents the coefficient index within the i^{th} subband. The M-band analysis and synthesis FB with quantization noise is described in Figure 1.

In [8] it has been shown that the adaptation process for designing a synthesis FB supposes to find a new reconstruction matrix:

$$\mathbf{G}' = \left[\mathbf{G}'_0, \ldots, \mathbf{G}'_{Q-1}\right], \quad Q \leq M,$$

such that the reconstructed signal, $\widetilde{x}(m) = \mathbf{G}' y_b(m)$, has the minimum MSE with respect to the original signal:

$$\mathbf{G}' = \min_{\mathbf{G}} \sum_m \|x(m) - \widetilde{x}(m)\|^2. \tag{1}$$

The optimal solution $\mathbf{G}'$ to the minimization problem in (1) is generally obtained by linear regression and verifies:

$$\mathbf{R}_{xy} = \mathbf{G}' \cdot \mathbf{R}_{yy} \tag{2}$$

where:

$$\mathbf{R}_{xy} = \sum_m x(m) y_b^T(m) \quad \text{and} \quad \mathbf{R}_{yy} = \sum_m y_b(m) y_b(m)^T. \tag{3}$$

Moreover, if the auto-correlation matrix of the quantized subbands, $\mathbf{R}_{yy}$, is invertible, the solution can be found as:

$$\mathbf{G}' = \mathbf{R}_{xy} \cdot {\mathbf{R}_{yy}}^{-1}. \tag{4}$$

For 2D separable block-based transforms, the images are divided into blocks of size $M \times M$ pixels. Let us denote by $\widetilde{X}(m_1, m_2)$ and $Y_b(m_1, m_2)$ the blocks at the position (m_1, m_2) in the reconstructed and residual coefficient frames, respectively. The 2D-separable inverse transform, given by two matrices $\mathbf{G}_h$ and $\mathbf{G}_v$ of size $M \times M$, is applied successively on the rows and columns of the residual frames as follows:

$$\widetilde{X}(m_1, m_2) = \mathbf{G}_v \cdot Y_b(m_1, m_2) \cdot \mathbf{G}_h^T. \tag{5}$$

Another method for obtaining a separable transform is to use iterative algorithms [10] for solving the non-linear system in (5). Among them, the Levenberg-Marquardt method [11, 12] has proven its efficiency for 2D separable transforms. In this optimization approach, the objective function to be minimized, $f : \mathbb{R}^n \mapsto \mathbb{R}$, $n = M \times M$, is defined by:

$$f(\mathbf{G}) = \sum_m \|\widetilde{x}(m, \mathbf{G}) - x(m)\|^2 \tag{6}$$

This method has method has the advantage of obtaining only one adapted inverse matrix. It can be advantageous when coding the side information that has to be sent to the decoder.

However, the in-loop optimization introduce a non-negligible complexity at the encoder side, as multiple iterations [8] are needed for the optimization of $\mathbf{G}$. In the sequel, we propose a new inverse matrix adaptation method, which computes the best $\mathbf{G}'$ for each frame, in a single iteration.

3. INVERSE TRANSFORM ADAPTATION METHOD

In the proposed optimization framework, the first step of the encoding process coincides with the standard H.264/AVC one. Therefore, in a first time, the frame is completely encoded using the classical H.264/AVC coding scheme, and the obtained block residues, $\mathbf{R} = \mathbf{X} - \mathbf{X}_p$ (e.g. differences between the original pixels, $\mathbf{X}$, and their predictions, $\mathbf{X}_p$), as well as the coefficients obtained after the inverse quantization, $\tilde{\mathbf{Y}}$, are stored in order to perform the adaptation (here $\mathbf{G}_{AVC}$ denotes the standard H.264 transform matrix).

In a second time, after the frame is completely encoded and using these parameters, a new inverse transform $\mathbf{G}'$ is calculated such that it minimizes the distortion between $\mathbf{R}$ and the reconstructed residual, $\hat{\mathbf{R}} = {\mathbf{G}'}^T \cdot \tilde{\mathbf{Y}} \cdot \mathbf{G}'$:

$$\mathbf{G}' = min \sum_m \left\| \hat{\mathbf{R}}(m, \mathbf{G}') - \mathbf{R} \right\|. \tag{7}$$

Note that in previous works [8], the adaptation process is done by minimizing (6) and therefore performing several iterations of rate-distortion (R-D) optimization for each new inverse transform. In this framework, the R-D optimization is performed only once, in the first step, on the H.264/AVC transform coefficients, $\mathbf{G}_{AVC} \cdot \mathbf{R} \cdot \mathbf{G}_{AVC}^T$, therefore outside the closed encoding loop. The new inverse transform matrix found in (7) can thus be seen as sub-optimal in its performance, given no further R-D optimization is performed. Indeed, the distortion minimization in (7) is done using in

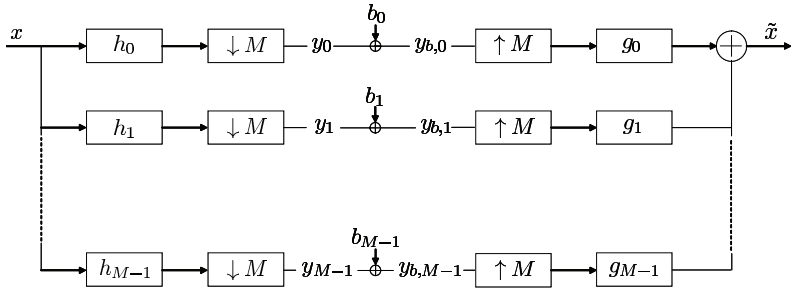

Fig. 1. M-band analysis and synthesis FB with quantization noise.

the reconstructed residual, $\hat{\mathbf{R}}$, the quantized coefficients obtained in the first step (e.g. standard H.264/AVC encoding). However, even sub-optimal, the proposed framework reduces considerably the complexity at the encoder side compared to the previously proposed adaptation scheme, since there is no longer the necessity of multiple R-D optimization passes.

In hybrid coders such as H.264/AVC, it is essential that both the encoder and the decoder have exactly the same version of the reconstructed frames. Since any block can be used as reference for future blocks, the smallest mismatch between the block reconstructions at encoder and decoder could generate an effect known as *drift*. Since the proposed adaptation is not done within the closed encoding loop, the inverse transform used at encoder, $\mathbf{G}_{AVC}$, and for which the block predictions have been obtained, differs from the matrix $\mathbf{G}'$ to be used at decoder side. As the drift effect is cumulative, i.e. the errors within a block propagate to the blocks referencing it, it is important to calculate at the decoder the residues reconstructed using the standard transform matrix, $\tilde{\mathbf{R}} = \mathbf{G}_{AVC}^T \cdot \tilde{\mathbf{Y}} \cdot \mathbf{G}_{AVC}$, and use them in the referencing process. Therefore, in this framework, the new adapted inverse transform matrix should be sent to the decoder as complement to the standard H.264/AVC bitstream (e.g. $\tilde{\mathbf{Y}}$) obtained in the first step.

At decoder side, the inverse transform will be performed twice for each block, in order to eliminate the drift. In a first step, the block is reconstructed using the standard inverse transform, $\mathbf{G}_{AVC}$:

$$\tilde{\mathbf{X}}_{AVC} = \mathbf{G}_{AVC}^T \cdot \tilde{\mathbf{Y}} \cdot \mathbf{G}_{AVC} + \mathbf{X}_p, \tag{8}$$

where $\mathbf{G}_{AVC}$ is defined as:

$$\mathbf{G}_{AVC} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix}. \tag{9}$$

As previously mentioned, this standard reconstruction $\tilde{\mathbf{X}}_{AVC}$ will be used in the reverse referencing process, in order to match exactly the block used by the standard H.264/AVC encoder in the prediction process. The second inverse transform, using the adapted matrix $\mathbf{G}'$, can be seen as an enhanced version of the first reconstruction and therefore gives the final decoded block:

$$\tilde{\mathbf{X}}' = \mathbf{G}'^T \cdot \tilde{\mathbf{Y}} \cdot \mathbf{G}' + \mathbf{X}_p. \tag{10}$$

The decoding block scheme is presented in Fig. 2. Note that the decoding process is entirely compatible with the standard one, i.e. $\tilde{\mathbf{X}}_{AVC}$ can be used in the absence of the adaptive decoding framework proposed in this work. Although the inloop deblocking filter is not illustrated in this diagram for simplicity, it considerably improves the coding performance, specially the quality of the inter predictions. It is used in the standard branch of the decoder (as well as in the coder). The adapted reconstruction also benefits from this technique.

In the adaptive inverse transform optimization framework $\mathbf{G}'$ has to be encoded and sent to the decoder side. The global performance of the proposed scheme depends therefore on the how efficiently this adaptation matrix is coded. The extra rate might considerably reduce or even cancel the coding gain obtained by the adaptation process, making thus the scheme unusefull, specially at lower resolutions. In this work, $\mathbf{G}'$ is encoded differentially using the standard H.264/AVC transform $\mathbf{G}_{AVC}$ as prediction, e.g. $\mathbf{r}_{\mathbf{G}'} = \mathbf{G}' - \mathbf{G}_{\mathbf{AVC}}$. The precision of the residual $\mathbf{r}_{\mathbf{G}'}$ is reduced by a simple scalar quantization. The quantization step has to be chosen carefully since it negatively impact the results (the quantized matrix no longer satisfies Eq. (7)), specially at lower bitrates (at higher bitrates the adapted inverse transform is closer to the standard one). The resulted quantized matrix is encoded as a 4×4 quantized coefficient block. Even though it was possible to integrate the coding of the adapted inverse transform into the syntax of H.264/AVC, it is sent separately to the decoder in order keep the encoded sequence standard compliant.

4. EXPERIMENTAL RESULTS

In order to evaluate the rate-distortion (R-D) performance of the proposed scheme, our adaptive framework has been implemented in JM 18.2 [1]. The adaptation was only performed on the 4×4 filter bank.

[1] http://iphome.hhi.de/suehring/tml/

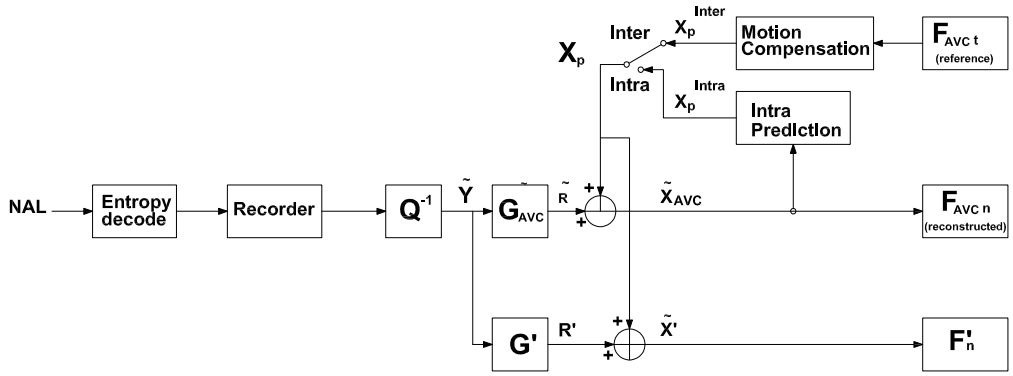

Fig. 2. Block diagram of the decoding process for the proposed inverse transform optimization scheme. [$\mathbf{F}_{AVCn}$ represents the n-th frame decoded using the standard H.264/AVC and available for being used as reference. $\mathbf{F}'_n$ represents the n-th frame decoded using the adapted inverse transform.]

The configuration parameters were chosen according to the recommendation [13]. In order to widen even more the range of bitrates, two additional points were also included, e.g. 42 and 47, for intra slices (43 and 48 for P Slices). These additional QPs allowed performance evaluation at considerably low bitrates. In the followings, we propose to analyse the results obtained for low bitrates (e.g. $QP \in \{32, 37, 42, 47\}$), medium bitrates (e.g. $QP \in \{27, 32, 37, 42\}$) and high bitrates (e.g. $QP \in \{22, 27, 32, 37\}$). The performance of the proposed scheme is presented using the Bjontegaard metric [14], both as average rate reduction for the considered QPs and PSNR gain difference, in dBs, with respect to the non-adaptive, standard implementation of H.264/AVC.

Table 1 presents the results obtained for several video sequences, namely the SD@30fps sequences City, Harbour and Soccer and the CIF@30fps sequences Coastguard, Mobile and Silent. In these tests, it was used a GOP size of 32 frames, in which the first frame is an I-frame, followed by P-frames. As the proposed adaptive method can be directly used in still image compression, Table 2 presents the results obtained for 3 512×512-pixels images, namely Barbara, Lena and Peppers.

Our adaptive coding framework presents gains for all tested images and video sequences, in all three bitrate intervals. It was observed that for P frames, at low bitrates, after the quantization, there are fewer non-zero coefficients that could benefit from the adaptation. Moreover, the impact of the adaptation matrix-rate is greater which can explain the smaller gains in this range of bitrate.

5. CONCLUSION

In this paper, we proposed an optimization framework for H.264/AVC, consisting in the adaptation of the inverse transform such that it minimizes the distortion of the decoded sequence in presence of quantization noise. Despite the fact that the inverse matrix adaptation is not performed the closed encoding loop, therefore it considers only the R-D optimization done with respect to the standard transform matrix, the proposed scheme proved its efficiency with respect to the standard H.264/AVC implementation for several bitrate intervals and sequences with different characteristics. Moreover, it considerably reduces the computational complexity at the encoder side compared with previously proposed adaptation schemes, since there is no longer the necessity of multiple R-D passes. The additional complexity at the decoder side (e.g. two inverse transforms are calculated in order avoid the drift, the remaining decoding process being kept identical) is considerably low compared to the previous complexity on the encoder side. Another advantage is that the resulting bitstream is entirely compatible with the standard H.264/AVC, if the new inverse transform matrix is ignored at decoder. Due to the low-complexity adaptive framework, as well as its efficiency for all tested bitrates, the proposed method can be successfully used in live encoding setups or, more generally, video streaming applications. Several extensions can further improve the results presented in this work. The adaptive scheme can be directly used in the high profile H.264/AVC, on 8×8 DCT-based transform or in bipredicted frames. At the encoder side, it would be possible to evaluate which would be the impact of sending the adapted matrix. The representation precision for the adapted matrix elements, therefore the rate used for sending it, could be decided frame by frame, therefore improving the performance especially at low bitrates.

6. REFERENCES

[1] P. P. Vaidyanathan, *Multirate systems and filter banks*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[2] K. Gosse and P. Duhamel, "Perfect reconstruction versus MMSE filter banks in source coding," *IEEE Transactions on Signal Processing*, vol. 45, no. 9, pp. 2188–2202, Sept 1997.

[3] N. Tizon and B. Pesquet-Popescu, "An adaptive synthesis filter bank for image decoding with fractional scala-

		Low bit-rates $Q_P \in \{32, 37, 42, 47\}$	Medium bit-rates $Q_P \in \{27, 32, 37, 42\}$	High bit-rates $Q_P \in \{22, 27, 32, 37\}$
City	Δ PSNR (dB)	-0.053	-0.023	0.010
(576×704)	Δ Rate (%)	0.039	-0.941	-2.246
Harbour	Δ PSNR (dB)	0.018	0.019	0.039
(576×704)	Δ Rate (%)	-0.467	-1.135	-1.817
Soccer	Δ PSNR (dB)	0.020	0.002	0.017
(576×704)	Δ Rate (%)	0.034	-0.572	-1.613
Coastguard	Δ PSNR (dB)	0.150	0.050	0.040
(288×352)	Δ Rate (%)	-2.114	-3.105	-4.250
Mobile	Δ PSNR (dB)	0.077	0.014	0.060
(288×352)	Δ Rate (%)	-3.118	-4.053	-5.691
Silent	Δ PSNR (dB)	0.104	0.085	0.051
(288×352)	Δ Rate (%)	3.854	1.584	-0.196

Table 1. Average gains obtained for several SD and CIF video sequences using Bjontegaard metric.

		Low bit-rates $Q_P \in \{32, 37, 42, 47\}$	Medium bit-rates $Q_P \in \{27, 32, 37, 42\}$	High bit-rates $Q_P \in \{22, 27, 32, 37\}$
Barbara	Δ PSNR (dB)	0.070	0.055	0.035
(512×512)	Δ Rate (%)	-1.446	-0.924	-0.585
Lena	Δ PSNR (dB)	0.065	0.059	0.037
(512×512)	Δ Rate (%)	-1.061	-0.963	-0.627
Peppers	Δ PSNR (dB)	0.062	0.035	0.026
(512×512)	Δ Rate (%)	-0.992	-0.523	-0.421

Table 2. Average gains obtained for several 512×512-pixels images using Bjontegaard metric.

bility," *IEEE 9th Workshop on Multimedia Signal Processing*, pp. 304–307, Oct 2007.

[4] Gregory K. Wallace, "The jpeg still picture compression standard," *Communications of the ACM*, pp. 30–44, 1991.

[5] G. Calvagno, G.A. Mian, R. Rinaldo, and W. Trabucco, "Two-dimensional separable filters for optimal reconstruction of JPEG-coded images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 7, pp. 777–787, Jul 2001.

[6] JVT of ISO/IEC and ITU-T VCEG, "Text description of joint model reference encoding methods and decoding concealment methods," *JVT-N046*, Jan 2005.

[7] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G.J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 688–703, Jul 2003.

[8] R. Galvao de Oliveira, M. Trocan, B. Pesquet, and N. Tizon, "An adaptive framework for h.264 optimized encoding," *In the proc. of 3rd IEEE European Workshop on Visual Information Processing (EUVIP)*, Paris, July 2011.

[9] Hari Kalva, "The h.264 video coding standard," *IEEE Multimedia*, vol. 13, pp. 86–90, 2006.

[10] James M. Ortega and Werner C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[11] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM Journal on Applied Mathematics*, vol. 11, pp. 431–441, 1963.

[12] K. Levenberg, "A method for the solution of certain problems in least squares," in *The Quarterly of Applied Mathematics*, 1944, vol. 2.

[13] T.K. Tan, G. Sullivan, and T. Wedi, "Recommended simulation common conditions for coding efficiency experiments, revision 1.," *Proceedings of VCEG Meeting ITU-T SG16 Q.6*, 2007.

[14] G Bjontegaard, "Calculation of average psnr differences between rd-curves," *13th VCEGM33 Meeting*, March 2001.