16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

A BELIEF-PROPAGATION BASED FAST INTRA CODING ALGORITHM FOR THE H.264/AVC FREXT CODER

Simone Milani

Dept. of Information Engineering, University of Padova
via Gradenigo, 6/B - 35131 Padova - Italy
e-mail: `simone.milani@dei.unipd.it`

ABSTRACT

The paper presents a complexity reduction strategy for the spatial Intra prediction of the H.264/AVC FRExt coder. The algorithm relies on selecting a reduced set of prediction modes according to their probabilities, which are estimated adopting a Belief-Propagation procedure. The resulting coding modes are then employed to find the optimal partitioning. Experimental results show that the proposed method permits saving up to 63% of the computational complexity required by an exhaustive rate-distortion optimization with a negligible loss in performance. Moreover, it guarantees an accurate control of the computational complexity with respect to other methods.

1. INTRODUCTION

Intra-only video coding is a widely used coding method in professional and surveillance video applications. This fact is partly due to its ease of editing and partly due to the significant amount of computational complexity required by motion estimation. In the H.264/AVC standardization process the compression performance of Intra coding was significantly improved by the adoption of spatial prediction. The pixels of the current block are predicted using the reconstructed pixels of neighboring blocks according to different orientations, which result closely related to the characteristics of the image correlation [1]. In the first version of the H.264/AVC standard, the spatial prediction is limited to either blocks of 4×4 pixels or whole macroblocks (MBs) of 16×16 pixels. In the FRExt extension of the standard, blocks of 8×8 pixels are considered too. As a consequence, the computational complexity of an exhaustive rate-distortion optimization is significantly increased because of the different partitioning modes and the number of prediction directions. In order to overcome this problem, a wide variety of complexity reduction strategies, together with the introduction of novel hardware accelerators, have been proposed in literature.

In [2] Pan *et al.* propose a fast Intra prediction algorithm that extracts the features of the images using Sobel edge operators and chooses the predictor according to their statistics. In a similar way, the approaches in [3] and [4] estimate the directional characteristics of each frame and use them to estimate the most probable prediction modes. The solution proposed in [5] evaluates the distortion produced by prediction in the transform domain, while temporal correlation existing between adjacent frames can be used too, as it is shown in [6].

Many approaches employ early-termination decision in order to reduce the computational complexity. This implies that the coding time depends on the characteristics of the processed video sequence (see [7] as an example where the relative reduction of coding time varies from 40% to 70%), and therefore, an *"a priori"* estimation of the computational complexity is not possible.

With respect to these strategies, the design of a complexity reduction algorithm that permits controlling the amount of required computation offers several advantages such as

- the possibility of adapting the algorithm to devices with different computational capabilities and power supply;
- an accurate estimation of the autonomy of mobile devices;
- the possibility of enabling *power save* configurations that gradually reduce the computational complexity (at the cost of a worse rate-distortion optimization) according to the remaining battery charge.

The solution proposed in this paper computes for each 4×4 prediction mode the probability that it minimizes the cost function. According to this probability mass function (pmf), the algorithm elects a limited set of modes (the most probable ones) as possible *"best-mode"* candidates and computes the cost function for each of them. The probability estimation is performed using a low-cost Belief-Propagation (BP) strategy that exploits the statistical dependence among adjacent blocks.

In the following, Section 2 will deal with the Intra prediction defined within the H.264/AVC FRExt standard, while Section 3 describes the BP approach adopted in the algorithm. Section 4 presents how the set of candidates is created, and Section 5 illustrates how the algorithm checks whether it is worth merging the blocks together or not. Experimental results reported in Section 6 show that the proposed algorithm performs well with respect to other solutions, and in addition, computational complexity can be controlled by increasing or decreasing the number of candidate modes. Final conclusions are drawn in Section 7.

2. THE INTRA PREDICTION IN THE H.264/AVC FREXT STANDARD

Since the beginning of the standardization process of the H.264/AVC codec, the Intra coding scheme has been characterized by block-based spatial prediction. The pixels in the current block are predicted from the neighboring ones according to a spatial predictor which is chosen among a standardized set of possible candidates. At first two Intra coding modes were defined, named `Intra4x4` and `Intra16x16` respectively. The first one performs spatial prediction on blocks of 4×4 pixels and has a set of 9 candidate predictors, while the second one predicts a whole macroblock of 16×16 pixels choosing one predictor among a set of 4. With

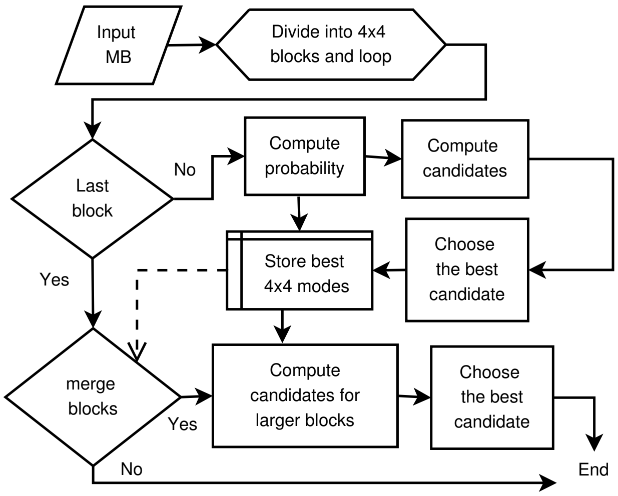


Figure 1: Block diagram of the proposed algorithm.

the extension of the coding standard (H.264/AVC FRExt), a novel `Intra8x8` mode was introduced using 9 possible candidates on blocks of 8×8 pixels [8].

Experimental results [1] have shown that the performance of spatial prediction coding in the H.264/AVC coder depends on the efficiency of the chosen directional predictor in modelling the characteristics of the signal. The default Intra coding method implemented in the reference H.264/AVC coder tests all the possible Intra prediction directions for each possible block partitioning (4×4, 8×8 and 16×16) and chooses the mode m that minimizes a Lagrangian cost function (see [9]). Since the best prediction mode m for the current 4×4 block is strongly correlated with the modes chosen for the spatially-neighboring blocks, in the H.264/AVC standard the bit rate $R(m)$ is coded after estimating the most probable prediction mode according to the modes of the upper and left blocks (see [8]). In our approach, we extend this idea estimating for each possible candidate mode the probability of being chosen as the best predictor.

Most of the fast Intra coding algorithms reduce the computational complexity by identifying the spatial orientation of the current block and selecting an appropriate set of candidate modes without performing a complete testing of all the possible predictors (see [2]). At the same time, these algorithms select the MB partitioning (`Intra4x4` or `Intra16x16`) that suits better to the current macroblock. As a consequence, it is possible to divide the implemented strategy into three phases: the estimation of the spatial orientation for the current block, the creation of an optimal set of candidates, and the detection of the best macroblock partitioning. In our approach, the orientation of the block correlation is found coding the current MB using the `Intra4x4` mode. Then, the 4×4 blocks are fused into either 8×8 blocks or a whole 16×16-pixels macroblock according to their prediction directions. Figure 1 reports the block diagram of the implemented algorithm, and the following sections will present the three phases in detail.

3. ESTIMATION OF THE ORIENTATION FOR 4×4 BLOCKS

Assuming that the array $\mathbf{p}(x,y) = [p_m(x,y)]$, $m = 0, \dots, M_0 - 1$, groups the probabilities $p_m(x,y)$ that the mode m is the best mode for the block at coordinates (x,y) (with M_0 the total number of candidate modes), it is possible to write the elements of $\mathbf{p}(x,y)$ as follows

$$p_m(x,y) = \mathbf{p}^T(x,y-1)\, Q^m(x,y)\, \mathbf{p}(x-1,y), \qquad (1)$$

where $Q^m(x,y) = [q^m_{i,j}(x,y)]$ is a $M_0 \times M_0$ matrix. The value $q^m_{i,j}(x,y)$ represents the conditional probability that mode m is the best mode for the current block at (x,y) given that i and j are the best modes for blocks at coordinates $(x,y-1)$ and $(x-1,y)$ respectively. The coding routine could select a reduced set $\mathscr{M}$ of M $(< M_0)$ possible candidates. This selection depends on the position of the block[1] and on the complexity reduction algorithm, which could avoid testing some prediction modes in order to reduce the computational load. As a consequence, Equation (1) can be modified as follows

$$\begin{aligned} \tilde{p}_m(x,y) \;=\; & \mathbf{p}^T(x,y-1)\, P^T_M(x,y-1)\, Q^m(x,y) \\ & P_M(x-1,y)\mathbf{p}(x-1,y) \end{aligned} \qquad (2)$$

where $P_M(x,y)$ is a singular projection matrix that sets to 0 some positions of $\mathbf{p}(x,y)$ according to which candidate modes are available. In this way, we obtain the array $\tilde{\mathbf{p}}(x,y) \neq \mathbf{p}(x,y)$ that could lead to a different set $\tilde{\mathscr{M}} \neq \mathscr{M}$ of candidate modes for the block at position (x,y). As a possible drawback, the chosen predictor could not match accurately the orientation of the local correlation either because the best mode is not included in the set $\tilde{\mathscr{M}}$ or because all the required neighboring pixels are not available and the most appropriate predictor can not be adopted. The finally chosen mode $\tilde{m}$ $(\neq m)$ could result sub-optimal for the current block and is going to affect the accuracy of probability estimation for the following adjacent blocks. It is possible to mitigate this effect by adopting a Belief-Propagation (BP) strategy that refines the statistics. Before coding the block at the coordinates (x,y), the BP procedure propagates the information about the best modes for the upper and left blocks, named $\tilde{m}(x,y-1)$ and $\tilde{m}(x-1,y)$. The coding routine estimates a probability distribution $\tilde{\mathbf{p}}(x,y)$ for the current block via equation (1), where

$$\begin{aligned} p_m(x-1,y) &= \begin{cases} 0 & m \neq \tilde{m}(x-1,y) \\ 1 & m = \tilde{m}(x-1,y) \end{cases} \\ p_m(x,y-1) &= \begin{cases} 0 & m \neq \tilde{m}(x,y-1) \\ 1 & m = \tilde{m}(x,y-1). \end{cases} \end{aligned} \qquad (3)$$

According to the values of $\tilde{\mathbf{p}}(x,y)$, all the possible prediction modes are sorted in decreasing probability order, and the most probable ones are included in the set $\tilde{\mathscr{M}}$ according to the criteria that will be described in Section 4. After finding the mode that minimizes the cost function among the candidates in $\tilde{\mathscr{M}}$, the BP approach propagates this result to the previously coded blocks in order to refine the accuracy of the estimated mode probability (i.e. $\tilde{\mathbf{p}}(x,y-1)$ and $\tilde{\mathbf{p}}(x-1,y)$). The vector $\tilde{\mathbf{p}}(x,y)$ of eq. (3) is replaced by a *"soft"* version (a likelihood) estimated via a reversed version of equation (1).

[1] The 4×4 blocks are coded according to a fixed order, and therefore, some prediction modes can not be adopted since the necessary reference pixels have not been coded yet.

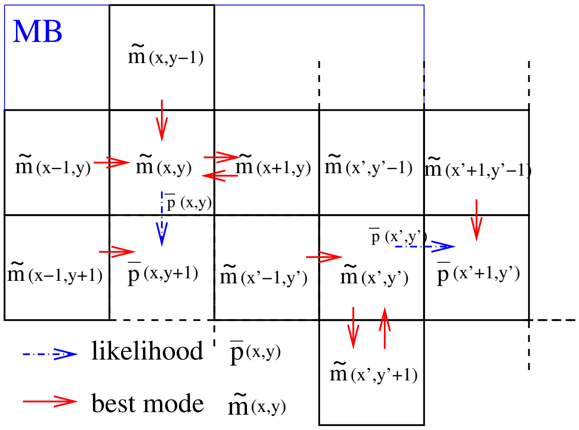


Figure 2: Probability propagation according to the implemented Belief Propagation approach.

Figure 2 reports a graphic example of the procedure. There, some messages propagates hard information (red arrows) regarding the chosen prediction modes, while others communicates likelihoods associated to the prediction mode of 4×4 blocks (blue arrows).

The new arrays $\tilde{\mathbf{p}}(x,y-1)$ and $\tilde{\mathbf{p}}(x-1,y)$ affect the estimated mode probability distribution for the following blocks and improve the compression performance of the fast Intra coding algorithm. Moreover, the best prediction modes found for `Intra4x4` coding are used to characterize the best-mode probability of prediction modes for bigger blocks in case the rate-distortion algorithm has chosen to merge the 4×4 blocks together, as described in Section 5.

After the optimization algorithm has chosen to merge together 4×4 blocks into 8×8 blocks, the coder estimates an `Intra8x8` best-mode probability distribution $\mathbf{p}^{8\times8}$ according to the previously found `Intra4x4` modes. The adopted approach estimates three different mode probability distributions $\mathbf{p}^{8\times8,i}$, $i=v,h,d$, which are dependent on the best Intra prediction modes of vertical, horizontal and diagonal couples of 4×4 blocks respectively (see Figure 3). Using the same notation of equation (1), it is possible to write $\mathbf{p}^{8\times8,i}=[p_m^{8\times8,i}]$, $i=v,h,d$ and $m=0,\ldots,8$, as

$$
\begin{aligned}
p_m^{8\times8,v} &= \tilde{\mathbf{p}}^T(x,y)\; F_m^{8\times8,v}\; \tilde{\mathbf{p}}(x,y+1)\\
p_m^{8\times8,h} &= \tilde{\mathbf{p}}^T(x,y)\; F_m^{8\times8,h}\; \tilde{\mathbf{p}}(x+1,y)\\
p_m^{8\times8,d} &= \tilde{\mathbf{p}}^T(x,y)\; F_m^{8\times8,d}\; \tilde{\mathbf{p}}(x+1,y+1)
\end{aligned}
\qquad (4)
$$

where $\tilde{\mathbf{p}}(x,y)$ represents the chosen prediction mode (see equation (3)) and $F_m^{8\times8,i}$, $i=v,h,d$, is the conditional probability matrix of 8×8 Intra predictor m given vertical, horizontal, and diagonal couples of 4×4 modes. In this way, `Intra8x8` best-mode probability estimation relies on the results of `Intra4x4` coding which has already be performed on the current macroblock.

4. ESTIMATION OF THE SET OF M CANDIDATES

4.1 Computation of the most probable prediction modes

After estimating the probability array $\tilde{\mathbf{p}}(x,y)$, the coding routine has to identify those modes that are more likely to be the best predictors for the current 4×4 block. The number of candidate modes M is usually set to the average value $\overline{M}$ but can vary according to the characteristics of the probability distribution identified by $\tilde{\mathbf{p}}(x,y)$. In fact, experimental data show that the entropies of distributions $\tilde{\mathbf{p}}(x,y)$ vary, and therefore, the mode probability distributions with a lower entropy only needs a reduced number of candidates.

Named $\tilde{\pi}(x,y)$ and ϕ the pmfs obtained reordering in decreasing probability order $\tilde{\mathbf{p}}(x,y)$ and the average best-mode probability distribution respectively, the value $\overline{M}$ is assumed to be the μ-th percentile for ϕ, and the parameter M is chosen such that it represents the μ-th percentile for $\tilde{\pi}(x,y)$.

The value $\overline{M}$ controls the average number of modes to be tested for each block and permits controlling the computational complexity. In this way it is possible to provide the same probability of finding the best prediction mode with a limited sets of candidates to all the 4×4 blocks of the image. This equalization permits saving some computational complexity without affecting the coding performance of the algorithm.

4.2 Mode reduction based on the dominant direction (DD algorithm)

According to the probability values of $\tilde{\mathbf{p}}(x,y)$, the M most probable modes are included in the set $\tilde{\mathscr{M}}$ of candidates. Whenever the entropy associated with $\tilde{\mathbf{p}}(x,y)$ is high, it is possible that the set $\tilde{\mathscr{M}}$ includes modes with orthogonal spatial orientations. Therefore, a further reduction of the candidate modes can be obtained by estimating whether horizontal or vertical modes are dominant in the distribution $\tilde{\mathbf{p}}(x,y)$. The number of orientations in the set $\tilde{\mathscr{M}}$ which are close to the vertical one is compared with the number of candidate modes which have a spatial orientation close to the horizontal one. In case one of them prevails, the other modes are deleted from the set $\tilde{\mathscr{M}}$.

4.3 Computation of the candidates for 8×8 blocks

A procedure similar to that of Subsection 4.1 is adopted in order to estimate the sets of candidate $\mathscr{M}_{8\times8}$ for the current 8×8 block from $\mathbf{p}^{8\times8,i}$, $i=v,h,d$. Each mode probability distribution $\mathbf{p}^{8\times8,i}$ infers a different set $\mathscr{M}_{8\times8,i}$, $i=v,h,d$, of candidate modes which is obtained in the same way of the set of M possible candidate modes for `Intra4x4` blocks. In case the set $\mathscr{M}_{8\times8}$ obtained from the intersection of the sets

$$
\mathscr{M}_{8\times8} = \mathscr{M}_{8\times8,v} \cap \mathscr{M}_{8\times8,h} \cap \mathscr{M}_{8\times8,d} \qquad (5)
$$

is not empty, the coding algorithm merges the 4×4 blocks into an 8×8 block and tests the predictors included in the set $\tilde{\mathscr{M}}$ looking for the one that minimizes the cost function.

As for the `Intra16x16` coding, all the 4 possible predictions are tested since the estimation of the best-mode probability for the 16×16 block from the best `Intra4x4` modes is not trivial.

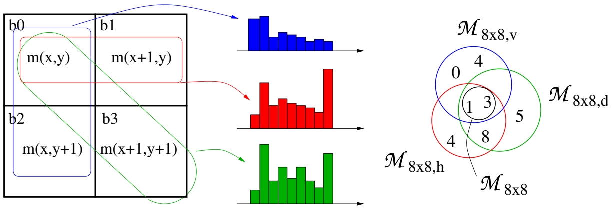


Figure 3: Merging operation for 4×4 blocks.

5. ESTIMATION OF MODE SIZE FOR INTRA PREDICTION

After finding the best mode for each 4×4 block in the current MB, the coding routine tests whether it is better to use bigger blocks. In a first step the algorithm checks whether it is possible to merge together the 4×4 blocks into blocks of 8×8 pixels. In case the orientations of each 4×4 block are the same or close, the merging results convenient with respect to the `Intra4x4` block partitioning since a reduced number of predictors needs to be coded in the transmitted bit stream. In order to detect these configurations, the encoder estimates the orientation differences $d(\tilde{m}(x,y), \tilde{m}(x+i, y+j))$, $i, j = 0, 1$, between vertical, horizontal and diagonal couples of 4×4 blocks (see Figure 3) within the current 8×8 block. If the average difference

$$\overline{d} = \frac{|d(\tilde{m}(x,y),\tilde{m}(x+1,y))|}{3} + \frac{|d(\tilde{m}(x,y),\tilde{m}(x,y+1))|}{3} + \frac{|d(\tilde{m}(x,y),\tilde{m}(x+1,y+1))|}{3} \qquad (6)$$

is lower than $40°$, the 4×4 blocks at (x,y), $(x+1,y)$, $(x,y+1)$, and $(x+1,y+1)$ could be merged into one block of 8×8 pixels. In case the condition on $\overline{d}$ is verified for all the 8×8, the `Intra8x8` coding mode is enabled. Moreover, the encoding routine tests whether it is worth merging the 8×8 blocks into one common 16×16 prediction block considering the `Intra4x4` modes for the blocks at the border of 8×8 blocks. In case the average absolute difference between the orientations of 4×4 blocks lying at the borders of 8×8 blocks is lower than $40°$ too, the `Intra16x16` prediction mode is chosen for the current macroblock. In this way, the wider block partitioning modes are tested only in case the orientations for the 4×4 blocks are approximately uniform, otherwise either 4×4 or 8×8 partitioning is preferred.

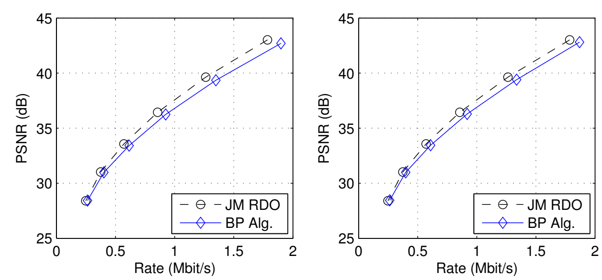
The whole fast intra coding algorithm can be summarized by the pseudocode reported by Algorithm 1: Experimental results will show that this choice leads to good performance with respect to other proposed solutions.

6. EXPERIMENTAL RESULTS

In order to test the efficiency of the presented algorithm, we coded different sequences with different quantization parameters and enabling different Intra coding modes. The proposed Intra coding strategy was implemented into the JM10.1 software. In our tests we adopted the same parameter setting used in [2], coding different sequences with $QP =$ 28, 32, 36, 40 and all Intra frames. At first we evaluated the performance of `Intra4x4` and `Intra16x16` modes only, comparing the computational complexity, the PSNR value, and the coded bit rate of the presented solution with those

Algorithm 1 Fast Intra coding procedure for a macroblock.

```
Test Intra4x4 coding mode
for each 4 × 4 block in the current macroblock do
    compute p̃(x,y)
    compute M and create the set 𝓜̃
    reduce 𝓜̃ via the DD algorithm (see Subsection 4.2)
    test all the modes in 𝓜̃ an
end for
check if it is worth merging the 4 × 4 blocks into bigger
blocks as described in the current Section
if Intra8x8 is to be enabled then
    for each 8 × 8 block in the current macroblock do
        compute p̃^{8×8,i}, i = v,h,d
        compute 𝓜_{8×8} and find the best mode
    end for
end if
if Intra16x16 mode is to be enabled then
    find the best prediction mode
end if
choose the MB Intra coding mode that minimize the total
cost function
```



(a) `silent` QCIF $\overline{M} = 5$ (b) `silent` QCIF $\overline{M} = 7$

Figure 4: PSNR vs. rate for Intra-only coded sequence `silent` with different target number $\overline{M}$ of candidates (`Intra4x4` only).

provided by the full-complexity rate-distortion optimization algorithm implemented in the reference software.

Experimental results show that the PSNR vs. rate curves of the two methods are quite close (see Figure 4). It is possible to notice that coding performance in terms of rate-distortion optimization is related to the target number $\overline{M}$ of candidate modes, which can vary according to the available computational resources or the remaining power supply.

Table 1 reports the average values of PSNR loss, together with the rate increment and the saved coding time, for different configurations of the algorithm (with `Intra8x8` mode disabled). The performance of the proposed approach is compared with the results of the algorithm in [2] (reported at the bottom of the table). The presented algorithm is able to reduce the coding time of approximately 62% with respect to the JM approach with an average rate increment lower than 5% and a PSNR loss of 0.15 dB ($\overline{M} = 5$). On the other hand, the rate increment is slightly lower for the approach of Pan *et al.* (3.47), but the computational complexity and the quality loss result higher. Since rate or quality increments are related to the setting of the adopted rate-distortion optimization strategy, the performance of the proposed algorithm is close to that of the approaches in [2] and [7], despite the complexity reduction does not significantly vary with respect to the in-

M	Sequence	Δ Bits (%)	Δ PSNR (dB)	Δ Time (%)
5	container (qcif)	5.37	−0.13	−62.81
	news (qcif)	5.32	−0.14	−62.41
	silent (qcif)	8.03	−0.15	−63.11
	coastguard (qcif)	3.06	−0.14	−63.02
	bus (cif)	2.62	−0.15	−62.33
	tempete (cif)	4.92	−0.21	−62.43
average		4.89	−0.15	−62.68
6	container (qcif)	5.48	−0.11	−55.26
	news (qcif)	5.00	−0.11	−54.37
	silent (qcif)	7.52	−0.10	−54.91
	coastguard (qcif)	3.16	−0.12	−55.92
	bus (cif)	2.61	−0.13	−54.59
	tempete (cif)	4.92	−0.17	−54.54
average		4.78	−0.12	−54.93
6 w/o DD	container (qcif)	5.64	−0.08	−50.43
	news (qcif)	4.96	−0.09	−47.79
	silent (qcif)	7.12	−0.06	−48.40
	coastguard (qcif)	3.64	−0.09	−51.10
	bus (cif)	2.73	−0.10	−49.41
	tempete (cif)	5.05	−0.13	−50.07
average		4.85	−0.09	−49.53
Pan *et al.* [2]	container (qcif)	3.69	−0.23	−56.36
	news (qcif)	3.90	−0.29	−55.34
	silent (qcif)	3.54	−0.18	−65.17
	coastguard (qcif)	2.36	−0.11	−55.03
	bus (cif)	3.85	−0.10	−58.12
	tempete (cif)	3.51	−0.23	−57.70
average		3.47	−0.19	−57.95

Table 1: Experimental results with `Intra8x8` disabled.

M	Sequence	Δ Bits (%)	Δ PSNR (dB)	Δ Time (%)
6	**container (qcif)**	5.54	−0.10	−61.81
	news (qcif)	5.47	−0.13	−60.28
	coastguard (qcif)	4.19	−0.16	−62.47
	bus (cif)	3.21	−0.14	−58.74
	tempete (cif)	5.07	−0.19	−57.29
average		5.14	−0.14	−59.25

Table 2: Experimental results with `Intra8x8` enabled.

Algorithm	E[ΔTime (%)]	range for ΔTime (%)
BP $\overline{M} = 5$	−62.68	$[-63.11, -62.33]$
BP $\overline{M} = 6$	−54.93	$[-55.92, -54.37]$
BP $\overline{M} = 7$ w/o DD	−49.53	$[-57.79, -51.110]$
Pan *et al.* [2]	−59.57	$[-65.38, -55.03]$
Yong-dong *et al.* [7]	−60.38	$[-68.70, -40.30]$

Table 3: Experimental results for different algorithms.

put sequence. The coding performance results slightly worse with respect to that of [4] despite the test setting is different (CAVLC is employed in place of CABAC) and the complexity saving of our approach is not significantly affected by the coded sequence.

Table 2 reports some experimental results obtained enabling the `Intra8x8` coding mode too. In this case the average performance does not significantly change, but the complexity reduction results slightly more variable because of the increased number of coding modes.

Table 3 reports the range of variation for the saved coding time of different fast Intra coding algorithms. It is possible to notice that the computational complexity does not significantly vary according to the input sequence and can be tuned according to the desired configuration.

7. CONCLUSIONS

The paper presented a fast Intra coding algorithm that is based on estimating for each spatial prediction mode the probability of being chosen as the best predictor. The probability estimates are obtained via Belief Propagation strategy that relies on the statistical dependence existing between spatially neighboring blocks. In a following step, the presented algorithm tries to identify the macroblock partitioning mode that better suits the current macroblock according to the coding results of the `Intra4x4` mode. Experimental results show that it is possible to obtain a significant saving in terms of coding time (approximately 61%) with a negligible decrement of the PSNR value and a small average increment (less than 5.14%) in the bit rate. Moreover, the presented strategy permits an accurate control on the encoding complexity, which does not significantly vary depending on the input video sequence and can be tuned according to the power supply level and to the available computational resources.

Acknowledgements

Part of the work was done in collaboration with Luca Celetto of STMicroelectronics, Agrate Brianza (MI), Italy.

REFERENCES

[1] L. Cappellari and G. A. Mian, "Analysis of joint predictive-transform coding for still image compression," *Signal Processing*, vol. 84, no. 11, pp. 2097–2114, Nov. 2004.

[2] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. G. Li, D. Wu, and S. Wu, "Fast mode decision algorithm for intraprediction in H.264/AVC video coding," *IEEE Trans. on CSVT*, vol. 15, no. 7, pp. 813–822, July 2005.

[3] J.-S. Ryu and E.-T. Kim, "Fast Intra coding method of H.264 for video surveillance system," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 7, no. 10, pp. 76–81, Oct. 2007.

[4] J. Kim, K. Jeon, and J. Jeong, "H.264 Intra Mode Decision for Reducing Complexity Using Directional Masks and Neighboring Modes," in *Proc. of PSIVT 2006*, Hsinchu, Taiwan, Dec. 11 – 13 2006, pp. 959–968.

[5] J. Xin, A. Vetro, and H. Sun, "Efficient macroblock coding-mode decision for H.264/AVC video coding," in *Proc. of PCS 2004*, San Francisco, CA, USA, Dec. 15 – 17 2004.

[6] J. Xin and A. Vetro, "Fast mode decision for Intra-only H.264/AVC coding," in *Proc. of PCS 2006*, Beijing, China, Apr. 24 – 26 2006.

[7] Z. Yong-dong, D. Feng, and L. Shou-xun, "Fast 4×4 Intra-prediction mode selection for H.264," in *Proc. of IEEE ICME 2004)*, Taipei, Taiwan, June 27 – 30 2004, pp. 1151–1154.

[8] T. Wiegand, "Version 3 of H.264/AVC," in *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)*, 12^{th} *Meeting*, Redmond, WA, USA, July 17 – 23 2004.

[9] G. J. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression," *IEEE Signal Processing Mag.*, pp. 74–90, Nov. 1998.