Fast H.264/AVC FRExt Intra Coding Using Belief Propagation

Simone Milani, *Member, IEEE*

***Abstract*—In the H.264/AVC FRExt coder, the coding performance of Intra coding significantly overcomes the previous still image coding standards, like JPEG2000, thanks to a massive use of spatial prediction. Unfortunately, the adoption of an extensive set of predictors induces a significant increase of the computational complexity required by the rate-distortion optimization routine. The paper presents a complexity reduction strategy that aims at reducing the computational load of the Intra coding with a small loss in the compression performance. The proposed algorithm relies on selecting a reduced set of prediction modes according to their probabilities, which are estimated adopting a Belief-Propagation procedure. Experimental results show that the proposed method permits saving up to 60% of the coding time required by an exhaustive rate-distortion optimization method with a negligible loss in performance. Moreover, it permits an accurate control of the computational complexity unlike other methods where the computational complexity depends on the coded sequence.**

***Index Terms*—fast Intra coding, video coding, H.264/AVC, belief propagation, rate-distortion optimization.**

I. Introduction

In the H.264/AVC standardization process the compression performance of Intra coding was significantly improved by the adoption of spatial prediction in Intra frames. The pixels of the current block are predicted using the reconstructed pixels of neighboring blocks interpolated along 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. This strategy, together with an improved rate-distortion optimization and an effective arithmetic coding engine, permits obtaining a higher compression gain with respect to the previous image coding standards like JPEG2000 (see [2]), but unfortunately, the additional operations, which are required to find the most appropriate spatial prediction and block size, significantly increase the final computational complexity. This drawback makes the adoption of H.264 Intra coding more troublesome for traditional Intra-only video coding such as video surveillance systems, where real-time video processing is required and the computational load must be constrained. 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 [3] Pan *et al.* propose a fast Intra prediction algorithm that extracts the image features using Sobel edge operators and chooses the predictor according to their statistics. In a similar way, the approaches in [4], [5] extract the directional features of each frame and use them to estimate the most probable prediction modes. The solutions proposed in [6], [7] evaluate the distortion produced by prediction in the transform domain, while in [8] Kim *et al.* jointly extract the features of each block from both pixels and transform coefficients. In addition, temporal correlation existing between adjacent frames can be used too, as it is shown in [9].

Many approaches employ early-termination decision in order to reduce the amount of computation [10]. This makes the computational complexity vary significantly according to the processed video sequence (see [11] as an example where the relative reduction of coding time varies from 40% to 70%), and therefore, an *"a priori"* estimation of the resulting cost is not possible. At the same time, the performance of the algorithm varies according to the coded sequence like in the case of [12] where a machine learning algorithm is used to select the best prediction mode among a reduced set of candidates.

With respect to these methods, the design of a complexity reduction strategy that permits controlling the amount of required computation provides 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 battery-powered coding devices;
- the possibility of enabling power saving configurations that gradually reduce the computational complexity (at the cost of a worse rate-distortion optimization) according to the remaining battery charge.

The solution presented in the following computes, for each 4×4 prediction mode, the probability that it minimizes the cost function with respect to the other ones. According to this probability distribution, 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 end, the algorithm checks whether it is worth merging the blocks together or not.

In the following, Section II will describe the Intra prediction process in the H.264/AVC FRExt coder and the related complexity problems. Then, Section III will present the proposed algorithm that estimates the best-mode probability for each prediction orientation from the previous coding results. The estimated probability distribution permits computing a reduced set of candidate modes as described in Section IV. Section V will illustrate how the algorithm chooses whether it is worth merging blocks together or not, and a final outline of the whole

Simone Milani is with the Dept. of Information Engineering, University of Padova, via Gradenigo 6/B, 35131 Padova, Italy. Phone: +39 049 827 7641 Fax: +39 049 827 7699. E-mail: simone.milani@dei.unipd.it.

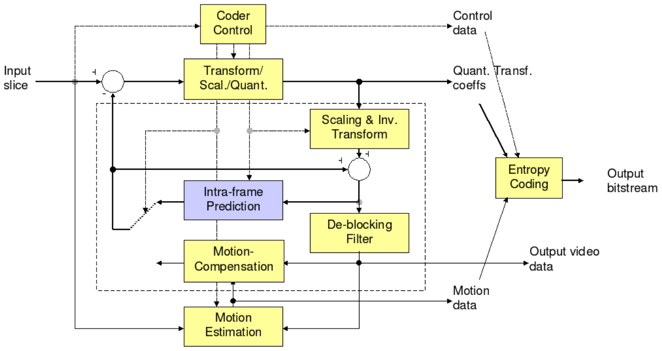


Fig. 1. Block diagram of the H.264/AVC coder.

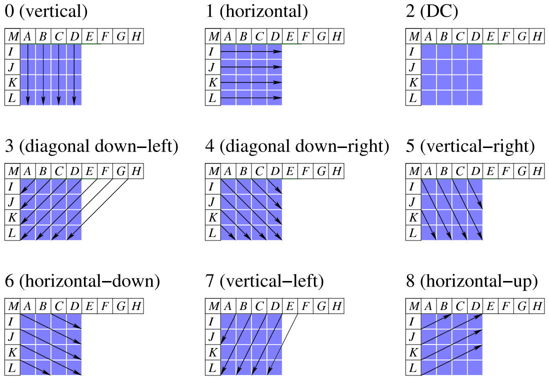


Fig. 2. Prediction modes of the luminance signal for the `Intra4x4` coding mode.

fast intra coding strategy will be provided. Experimental results, presented in Section VI, will show that the performance of the algorithm compares well with other solutions, and in addition, that the computational complexity can be controlled by increasing or decreasing the number of candidate modes. Final conclusions will be drawn in Section VII.

II. The Intra prediction in the H.264/AVC FRExt standard

Since the earliest steps of its standardization process, the Intra coding mode of the H.264/AVC codec has been characterized by block-based spatial prediction. Being one of the innovative elements introduced in the H.264/AVC standard with respect to the previous codecs, the standardization process has developed this feature in order to make it sufficiently flexible in characterizing the input image signal and improving the compression gain of Intra coding. In fact, most of the gain of H.264/AVC Intra mode over the previous image coding standards [2] is strictly related to the ability of the codec in characterizing the correlation structure of the coded image [1]. To this purpose, the Intra coding modes defined within the H.264/AVC standard estimate the orientation of spatial correlation that interlies between adjacent pixels and predict a whole block of pixels interpolating those lying on the borders of the block.

After a predictor block has been generated, the prediction residual block obtained subtracting the predictor from the original one is transformed (using the approximated DCT transforms defined within the H.264/AVC standard), and the resulting coefficients are quantized. The quantized coefficients are coded into a binary stream which is transmitted to the decoder, together with the other syntax elements of H.264/AVC. Figure 1 shows a block diagram of the structure of the H.264/AVC coder where the spatial prediction unit is evidenced. At the decoder, the coefficients are then dequantized and inversely-transformed in order to obtain the coded residual signal that, together with the predictor block, permits reconstructing the current signal. As a matter of fact, the predictor has to be created using the reconstructed pixels of the already-decoded neighboring blocks since the other pixel values are not available at the decoder. In addition, the encoder has to signal effectively in the coded bit stream how to generate the predictor block.

To this purpose, the standard defines a finite set of possible orientations, which correspond to a finite set of interpolating equations [13]. Each equation generates a possible candidate predictor processing the neighboring pixels lying along the upper and the left borders of the current block (which have already been decoded). The H.264/AVC codec selects the prediction block that characterizes most appropriately the local signal and permits obtaining a good compromise between rate and distortion. In fact, the chosen predictor must both minimize the distortion resulting from the prediction and limit the increment of the bit stream due to the coding of prediction modes and quantized coefficients.

In order to provide a certain degree of flexibility, block size can vary to adapt the prediction scheme to the statistics of the signal. For signals smooth on a wide region (i.e. regular backgrounds, homogeneous areas), bigger blocks are used since a large set of neighboring pixels result correlated among each other and it is possible to specify a single predictor for a wide partition of the image. In this way it is possible to reduce the amount of bits required to specify the adopted predictor. Whenever the image is highly irregular and the statistics of the spatial correlation frequently change, smaller blocks are employed in order to fit the chosen predictor to the characteristics of the signal. In this case, a greater number of bits is required to code the prediction orientations, but the amount of bits that code the transform coefficients decreases since the energy of the residual signal (and, as a consequence, the number of transform coefficients to be coded) is reduced.

During the standardization process of H.264/AVC, two Intra coding modes were at first defined, named `Intra4x4` and `Intra16x16` respectively. With regards to the luminance component of the video signal, the first coding mode performs spatial prediction on blocks of 4×4 pixels and has a set of 9 candidate predictors (see Fig. 2), while the second one predicts a whole macroblock of 16×16 pixels choosing one predictor among a set of 4. As for the chrominances, the spatial prediction is performed on blocks of 8×8 pixels using 4 prediction modes similar to those for luminance in `Intra16x16` mode. With the extension of the coding standard (H.264/AVC FRExt), a novel `Intra8x8` mode was introduced for luminance using 9 possible candidates on

$$\begin{aligned}\tilde{p}_m(x,y) &= \mathbf{p}^T(x,y-1)\ P_M^T(x,y-1)\ Q^m(x,y)P_M(x-1,y)\mathbf{p}(x-1,y) \\ &= \tilde{\mathbf{p}}^T(x,y-1)\ Q^m(x,y)\ \tilde{\mathbf{p}}(x-1,y)\end{aligned} \tag{3}$$

blocks of 8×8 pixels [13]. In these cases, the traditional 4×4 integer transform defined within the H.264/AVC standard is replaced by an approximate 8×8 integer DCT transform that fits the block size.

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 $L(m) = SAD(m) + \lambda\ R(m)$. The value $R(m)$ is the bit rate needed to code the current mode m in the bit stream, $SAD(m)$ is the distortion metric (Sum of Absolute Differences), and λ is a Lagrange multiplier that weights the influence of both distortion and bit rate in the cost function (see [14]).

In order to reduce the computational complexity required by the exhaustive Lagrangian approach, several fast intra coding solutions have been proposed in literature. Many algorithms reduce the computational complexity by identifying the local 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 [3]). At the same time, these algorithms analyze the spatial orientation field to select the most appropriate MB partitioning (`Intra4x4`, `Intra8x8`, or `Intra16x16`). Whenever the spatial orientations prove to be quite homogeneous within a certain region, it is possible to adopt larger prediction blocks.

As a consequence, in most of the fast intra coding strategies it is possible to identify the following three phases: the estimation of the local spatial orientation, the creation of an optimal set of candidate predictors (just one for the more aggressive algorithms), and the detection of the best macroblock partitioning.

In the following, the proposed approach will be presented by describing these three phases in detail.

III. Estimation of block orientations

In the proposed approach, the orientation of the image correlation is found coding the current MB using the `Intra4x4` mode and the spatial correlation between the best prediction mode of neighboring 4×4 blocks.

The best prediction mode m for the current 4×4 block is strongly correlated with the modes chosen for the spatially-neighboring blocks. This is utterly confirmed by the fact that 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 [13]). In the proposed approach, this idea is extended estimating for each possible candidate mode the probability of being chosen as the best predictor.

Assuming that the $M_0\times1$ array $\mathbf{p}(x,y) = [p_m(x,y)]$ ($m = 0,\ldots,M_0-1$) 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

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

where $Q^m(x,y) = [q^m_{i,j}(x,y)]$ is an $M_0\times M_0$ 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. However, it may happen that only a smaller set $\mathcal{M}$ of M candidate modes ($M < M_0$) are available for the block at (x,y), and therefore, the probabilities $p_{m'}(x,y)$ are 0 with $m' \notin \mathcal{M}$. This is the case of blocks placed at positions where some reference pixels are not available because of the frame boundaries or the block coding order (e.g. upper-right pixels can not be used since the corresponding neighboring block has not been coded yet). The same candidate modes reduction is found for all the blocks whenever the H.264/AVC coder adopts a fast intra prediction algorithm that tests only a selected set of candidates to constrain the computational complexity. This candidate modes reduction affects the best-mode probability array, which can be replaced with the relation

$$\tilde{\mathbf{p}}(x,y) = \ P_M(x,y)\ \mathbf{p}(x,y) \tag{2}$$

where $P_M(x,y)$ is a singular projection matrix that sets to 0 some elements of $\mathbf{p}(x,y)$ according to which candidate modes are available.

As a consequence, the best-mode statistics for the current block at position (x,y) can be estimated propagating the best-mode probability of previous blocks via equation (3) (which is a modified version of eq. (1)), and projecting the array $\mathbf{p}(x,y)$ onto the subspace of allowed modes using equation (2). The resulting array $\tilde{\mathbf{p}}(x,y)$ differs from the original estimate $\mathbf{p}(x,y)$ of eq. (1) because of the approximation introduced by the projection and leads to a different set $\tilde{\mathcal{M}} \neq \mathcal{M}$ of candidate modes. As a possible drawback, the chosen predictor could not match accurately the orientation of the local correlation either because the optimal mode is not included in the set $\tilde{\mathcal{M}}$ or because all the required neighboring pixels are not available and the most appropriate predictor can not be adopted. The finally chosen mode $\tilde{m}$ 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 for each block.

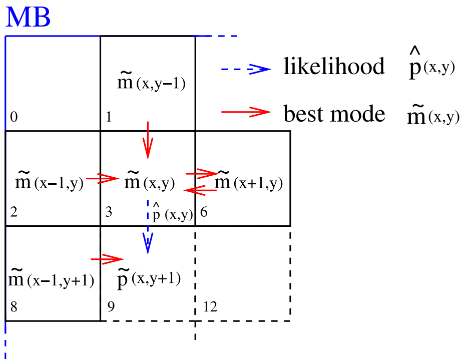


Fig. 3. Probability propagation according to the implemented Belief Propagation approach. Some message passing propagates hard information (solid red arrows) regarding the chosen prediction modes, while others communicates likelihoods associated to the prediction mode of 4×4 blocks (dashed blue arrows).

A. The Belief-Propagation procedure for spatial orientations of 4×4 blocks

Before coding the block at the coordinates (x, y), the mode estimation routine propagates through a BP procedure the information about the best modes for the upper and left blocks found during the coding operations (see Figure 3 for a graphic example). These modes are denoted here with $\tilde{m}(x, y-1)$ and $\tilde{m}(x-1, y)$ respectively. According to this, the coding routine estimates a probability mass function (pmf) $\tilde{\mathbf{p}}(x, y)$ for the current block via equation (3), where

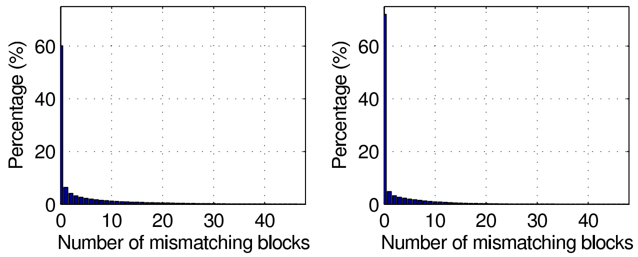
$$\tilde{p}_m(x-1,y) = \begin{cases} 0 & m \neq \tilde{m}(x-1,y) \\ 1 & m = \tilde{m}(x-1,y) \end{cases}$$
$$\tilde{p}_m(x,y-1) = \begin{cases} 0 & m \neq \tilde{m}(x,y-1) \\ 1 & m = \tilde{m}(x,y-1). \end{cases} \quad (4)$$

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 $\mathcal{M}$ according to the criteria that will be described in Section IV. After finding the mode that minimizes the cost function among the candidates in $\mathcal{M}$, 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 array $\tilde{\mathbf{p}}(x, y)$, whose elements are reported in eq. (4), is replaced by a *"soft"* best mode estimation $\hat{\mathbf{p}}(x, y)$ (a likelihood) computed using a reversed version of equation (1)

$$\hat{\mathbf{p}}(x,y) = \tilde{\mathbf{p}}^T(x,y-1)\ Q^{m,r}(x,y)\ \tilde{\mathbf{p}}(x+1,y). \quad (5)$$

The new arrays $\hat{\mathbf{p}}(x, y)$ affect the estimated mode probability distribution for the following blocks and improve the compression performance of the fast Intra coding algorithm. As an example, the elements for the probability array $\tilde{\mathbf{p}}(x, y+1)$ of block 9 in Fig. 3 are obtained via eq. (3) replacing the array $\tilde{\mathbf{p}}(x, y-1)$ with $\hat{\mathbf{p}}(x, y-1)$.

Experimental results have proved that the refinement step performed using equation (5) does not change the arrays $\tilde{\mathbf{p}}(x, y)$ in such a way that the order of candidate modes is altered. However, the likelihood estimate for the prediction mode $\tilde{m}(x, y)$ proves to be significant in the computation of the number of candidate modes as it will be explained in Subsection IV-A. Moreover, the best prediction modes found for `Intra4x4` coding are used to characterize the best-mode probability of predictions for bigger blocks in case the the rate-distortion algorithm has chosen to merge the 4×4 blocks together, as it will be described in Section V.



(a)

(b)

Fig. 4. Percentages of error propagation runs for 4×4 blocks of the first 15 frames for different sequences (CIF format coded with $QP = 28$ and $M = 4$). a) `news` b) `bus`.

Experimental data have also shown that the adopted BP solution proves to be robust with respect to mismatches between the best prediction mode found by the proposed algorithm and the actual best prediction mode found by the full-search strategy. Fig. 4 reports the percentages of blocks with a mismatching prediction mode interlying between two blocks with a matching prediction mode (i.e. the "runs" of mismatching blocks). The data have been obtained from the first 15 frames of the sequences `news` and `bus` (CIF format). The reported results show that for nearly 96% of blocks the mismatch of the prediction mode does not propagate farther than the neighboring macroblocks since the BP strategy permits correcting errors in the estimation of pmfs.

B. Probability estimation using a Finite State Machine

Standard probability estimation requires a significant number of floating point operations since the probability arrays $\tilde{\mathbf{p}}(x, y)$ and $\hat{\mathbf{p}}(x, y)$ have to be computed and updated after coding each block. Since the final aim of the proposed approach is to reduce the overall complexity of the Intra coding process, it is necessary to avoid an increment of the complexity for the estimation of the local orientation.

In order to overcome this problem, the array $\tilde{\mathbf{p}}(x, y)$ can be approximated choosing a probability distribution array $\check{\mathbf{p}}$ from a finite set $\mathcal{P}$ of 64 pmfs. Since the alphabet of probability arrays is limited, it is possible to characterize the probability distribution with an integer index. The set of distributions must be sufficiently exhaustive to include an approximated probability array close enough to the real one.

As a matter of fact, the cardinality $\|\mathcal{P}\|$ was chosen evaluating the efficiency of the probability estimation (in terms of compression efficiency at equal computational complexity) for different cardinalities. A lower number of pmfs brings to a poorer rate-distortion performance since the approximation of the real pmf $\tilde{\mathbf{p}}(x, y)$ leads to a coarse estimation of the best-mode probability for each prediction orientation, and as a consequence, to a wrong sorting of the prediction modes. The

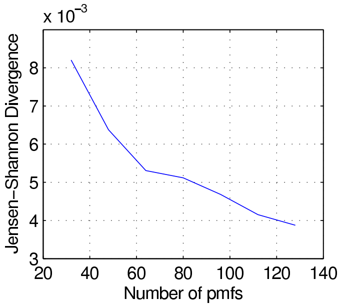


Fig. 5. Average Jensen-Shannon divergence for different cardinalities of the alphabet $\mathcal{P}$ (number of pmfs).

sorted array of modes associated to $\check{\mathbf{p}}$ differs from that associated to $\tilde{\mathbf{p}}(x, y)$ in the last positions since the most probable modes are usually correctly identified. As a consequence, the effects of approximation are mainly evident at high complexity configurations, where a great number of prediction modes are included in $\mathcal{M}$ comprising the wrongly-sorted ones. On the other hand, an excessive number of approximated pmfs in $\mathcal{P}$ implies an ineffective increment of the memory area required to store all the data.

In the proposed implementation, the alphabet $\mathcal{P}$ was generated by estimating an extensive set of probability arrays from many heterogeneous sequences and constructing an LBG quantizer with 64 cells on this set [15]. Sequences `foreman`, `news`, `mobile`, `teeny`, `salesman` (QCIF format) were coded with QP varying within the range $[20, 40]$ to generate the pmfs that permits building $\mathcal{P}$. The LBG cells were created via an iterative classification, where the distortion metric to minimize was the Jensen-Shannon divergence (JSD) between $\tilde{\mathbf{p}}(x, y)$ and $\check{\mathbf{p}} \in \mathcal{P}$

$$JSD\left(\tilde{\mathbf{p}}(x, y) \| \check{\mathbf{p}}\right) = \frac{1}{2} D\left(\tilde{\mathbf{p}}(x, y) \| \check{\mathbf{p}}\right) + \frac{1}{2} D\left(\check{\mathbf{p}} \| \tilde{\mathbf{p}}(x, y)\right) \quad (6)$$

with $D\left(\tilde{\mathbf{p}}(x, y) \| \check{\mathbf{p}}\right)$ the Kullback-Leibler divergence. After measuring the average Jensen-Shannon divergence associated to $\mathcal{P}$ with different cardinalities, we chose using 64 pmfs since it permits obtaining a good trade-off between the average JSD and the required memory space (see Fig. 5 where the average JSD is reported for different alphabets).

The conditional probability matrix $Q_m(x, y)$ is a linear combination of arrays $\check{\mathbf{p}} \in \mathcal{P}$, and the probability array $\tilde{\mathbf{p}}(x, y)$ at position (x, y) is updated after each iteration of the BP procedure using a Finite State Machine (FSM), where each state is related to an element of $\mathcal{P}$. In this way, it is possible to obtain an adaptive estimation of the probability for each prediction mode with limited computational complexity and memory area.

In the adopted approximation, each array $\check{\mathbf{p}}$ can be associated to an integer index, and the conditioned probability matrix $Q^m(x, y)$ has been fixed for the sake of complexity. As a matter of fact, $Q^m(x, y)$ is implemented using a look-up table that identifies a new state $\check{\mathbf{p}}$ in the FSM according to the states associated to the upper and the left blocks. This update process is quite similar to the probability update adopted in the CABAC coding engine of the H.264/AVC standard [16], made exception for the fact that the new $\check{\mathbf{p}}$ depends on two previous states of the FSM. Similarly, another fixed look-up table is adopted for the matrix $Q^{m,r}(x, y)$. The mapping functions implemented by the look-up tables have been estimated computing the conditioned probabilities between prediction modes obtained from an extensive set of coded training sequences. These probabilities have been mapped into states $\check{\mathbf{p}}$ according to the states of the upper and left blocks.

Experimental results will show that the implemented procedure permits reducing significantly the computational complexity while obtaining a good rate-distortion performance (see Section VI).

C. Estimation of probable spatial orientations for 8×8 blocks

As it will be shown in Section V, the best prediction modes for `Intra4x4` coding are used to estimate whether 8×8 blocks are more effective for spatial prediction. In case the `Intra8x8` mode is chosen, the fast intra prediction algorithm has to estimate the most probable prediction modes for 8×8 luminance blocks within the current macroblock.

After the coder has found the best prediction mode for each 4×4 block within the current 8×8 partition, the coder estimates an `Intra8x8` best-mode probability distribution $\mathbf{p}^{8\times 8}$. The adopted approach considers three different mode probability distributions $\mathbf{p}^{8\,\times\,8,\,i}$, $i = v, h, d$, which are generated from the best Intra prediction modes of vertical, horizontal and diagonal couples of 4×4 blocks, respectively (see Fig. 6). Using the same notation of equation (1), it is possible to write $\mathbf{p}^{8\times 8,i} = [p_m^{8\times 8,i}]$, $i = v, h, d$ and $m = 0, \ldots, 8$, as

$$\begin{aligned} p_m^{8\times 8,v} &= \tilde{\mathbf{p}}^T(x, y)\ F_m^{8\times 8,v}\ \tilde{\mathbf{p}}(x, y+1) \\ p_m^{8\times 8,h} &= \tilde{\mathbf{p}}^T(x, y)\ F_m^{8\times 8,h}\ \tilde{\mathbf{p}}(x+1, y) \\ p_m^{8\times 8,d} &= \tilde{\mathbf{p}}^T(x, y)\ F_m^{8\times 8,d}\ \tilde{\mathbf{p}}(x+1, y+1) \end{aligned} \quad (7)$$

where $\tilde{\mathbf{p}}(x, y)$ represents the chosen prediction mode (as equation (4) reports) and $F_m^{8\times 8,i}$, $i = v, h, d$, is the conditional probability matrix of 8×8 Intra prediction mode m given the 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 been performed on the current macroblock.

The following section will describe how probability distributions are used to estimate the set of candidate modes.

IV. Estimation of the set of M candidates

A. 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 prediction mode 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 distribution with a lower entropy only needs a reduced number of candidates so that the probability of finding the best mode $m(x, y)$ is higher than a certain threshold.

Named $\pi = [\pi_{\mathbf{m}}]$ the average mode probability array, M is chosen in such a way that

$$\begin{array}{l} \sum_{m=0}^{M-1} S_m(\tilde{\mathbf{p}}(x,y)) \;\leq\; \sum_{m=0}^{\overline{M}-1} S_m\left(\pi\right) \;\text{ and } \\ \sum_{m=0}^{M} S_m(\tilde{\mathbf{p}}(x,y)) \;>\; \sum_{m=0}^{\overline{M}-1} S_m\left(\pi\right) \end{array} \tag{8}$$

where $S_m(\cdot) : [0,1]^9 \rightarrow [0,1]$ is an ordering function that returns the m-th value of the input array in decreasing order. In statistical terms, assuming that $\overline{M}$ is the μ-th order percentile for the distribution π, the parameter M is the percentile of the same order for distribution $\tilde{\mathbf{p}}(x, y)$. The value $\overline{M}$ reports 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 set 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.

B. Further reduction of the possible candidates (DD algorithm)

According to the probability values of $\tilde{\mathbf{p}}(x, y)$, the M most probable modes are included in the set $\mathcal{M}$ of candidates. Whenever the entropy associated with $\tilde{\mathbf{p}}(x, y)$ is high, it is possible that the set $\mathcal{M}$ includes modes with orthogonal spatial orientations.

This fact is mainly due to the transient period in the probability estimation process, which may require several iterations before converging to an accurate estimate of mode statistics. It is also possible that spatial correlation in the coded region presents frequently-varying orientations, and therefore, the best prediction mode can significantly change from one block to another. In this case, in the probability array $\tilde{\mathbf{p}}(x, y)$, modes related to orthogonal directions present a comparable best mode probability, and therefore, they could be included together in the set $\mathcal{M}$.

A further reduction of the candidate modes can be obtained by estimating whether horizontal or vertical modes are dominant in the distribution $\tilde{\mathbf{p}}(x, y)$ and eliminating the dominated modes (Dominated Deletion - DD). The number of orientations in the set $\mathcal{M}$ 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 modes of the other type are deleted from the set $\mathcal{M}$.

However, this elimination procedure has to be constrained in order to avoid an excessive reduction of the candidate sets whenever the statistics of prediction orientations is not clearly biased on either vertical or horizontal directions. In order to avoid the deletion of probable candidate modes, the DD algorithm is performed only for modes greater or equal to 4 whenever the number of dominated modes is greater than a certain threshold value T.

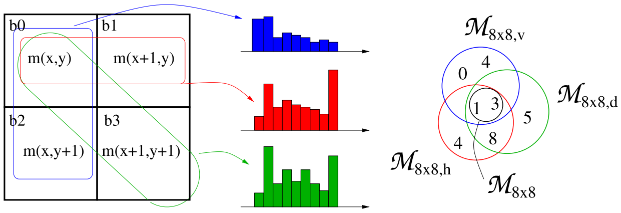


Fig. 6. Merging operation for 4×4 blocks.

In fact, modes vertical, horizontal, DC and diagonal prove to be the most probable ones since they can approximate well all the directions of the spatial correlation. The DD algorithm avoids deleting them from the candidate set in order to permit a change in the orientation of the predictor whenever the best prediction direction significantly varies for different blocks. The threshold T permits modulating the effects of the DD algorithm on the set $\mathcal{M}$, adding an extra degree of freedom in the modulation of the computational complexity.

Experimental results reported in Section VI will underline how the DD strategy permits a finer modulation of the computational complexity with respect to the parameter M.

C. Computation of the candidates for 8×8 and 16×16 blocks

In case the coding mode `Intra8x8` is enabled and the optimization routine has chosen to partition the current macroblock into 8×8 blocks (see Section V), the fast intra prediction algorithm has to estimate the most appropriate prediction modes from the results of `Intra4x4` mode. The transcoding algorithm reported in [17] chooses the most frequently used prediction direction, but in case the estimated 4×4 orientations prove to be nonuniform within the same 8×8 block, a better performance can be obtained testing a set $\mathcal{M}_{8\times8}$ of different candidates. A procedure similar to that of Subsection IV-A is adopted in order to estimate the sets of candidate $\mathcal{M}_{8\times8}$ 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 $\mathcal{M}_{8\times8,i}$, $i = v, h, d$, 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 $\mathcal{M}_{8\times8}$ obtained from the intersection of the sets

$$\mathcal{M}_{8\times8} = \mathcal{M}_{8\times8,v} \cap \mathcal{M}_{8\times8,h} \cap \mathcal{M}_{8\times8,d} \tag{9}$$

is not empty, the coding algorithm tests the predictors included in $\mathcal{M}_{8\times8}$ looking for the one that minimizes the cost function.

As for the `Intra16x16` coding, all the four 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. As for the chrominance components U and V, the same strategy is applied.

V. Estimation of best macroblock partitioning for Intra prediction

In the proposed approach, the orientation of the image 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 orientations.

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 4×4 blocks are the same or close, the merging of separate blocks 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 $d(\tilde{m}(x, y), \tilde{m}(x + 1, y))$, $d(\tilde{m}(x, y), \tilde{m}(x, y + 1))$, and $d(\tilde{m}(x, y), \tilde{m}(x + 1, y + 1))$ between vertical, horizontal and diagonal couples of 4×4 blocks within the current 8×8 block. The metric $d(\tilde{m}(x, y), \tilde{m}(x', y'))$ is computed as follows

$$d(\tilde{m}(x, y), \tilde{m}(x', y')) = |\angle\tilde{m}(x, y) - \angle\tilde{m}(x', y')| \qquad (10)$$

where $\angle m$ denotes the vertical angle associated to the spatial orientation of mode m. If the average difference

$$\begin{aligned} \overline{d} \quad &= \frac{d(\tilde{m}(x, y), \tilde{m}(x + 1, y))}{3} + \frac{d(\tilde{m}(x, y), \tilde{m}(x, y + 1))}{3} \\ &+ \frac{d(\tilde{m}(x, y), \tilde{m}(x + 1, y + 1))}{3} \end{aligned} \qquad (11)$$

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 blocks, the `Intra8x8` coding mode is enabled. The threshold value of 40° was found considering that it is approximately equal to 1.7 times the angle between two adjacent prediction orientations within the set of 9 predictors (which is approximately 22.5° [3]). Experimental results also proved that this choice is effective in identifying whether the merging is worth.

Moreover, the encoding routine tests whether it is worth including 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°, 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 procedure is depicted in the block diagram of Fig. 7. Experimental results will show that this choice leads to good performance with respect to other proposed solutions.

VI. Experimental results

In order to test the efficiency of the presented algorithm, different sequences were coded with different quantization parameter values and enabling different Intra coding modes. The proposed Intra coding strategy was implemented into the JM10.1 software. In the tests the adopted parameter setting is the same of [3], coding different sequences with only Intra frames and $QP = \{20, 24, 28, 32, 36, 40\}$. At first the performance of `Intra4x4` and `Intra16x16` modes only was evaluated, comparing the computational complexity, the PSNR value, and the coded bit rate of the presented solution with those provided by the full-complexity rate-distortion optimization algorithm implemented in the reference software. Experimental data show that the PSNR vs. rate curves of the two methods are quite close (see Figure 8). Moreover, the data reported in Fig. 8 show that the proposed approach compares well with respect to the algorithm in [3].

Table I reports the PSNR loss, together with the rate increment and the complexity reduction, for the proposed approach

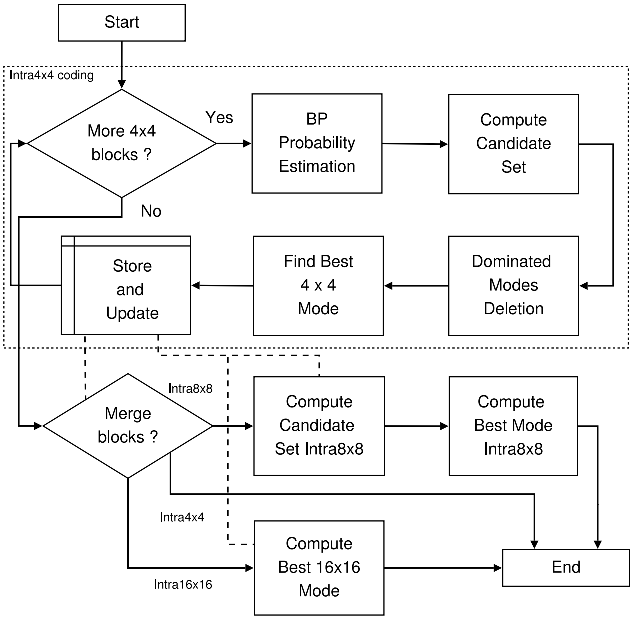
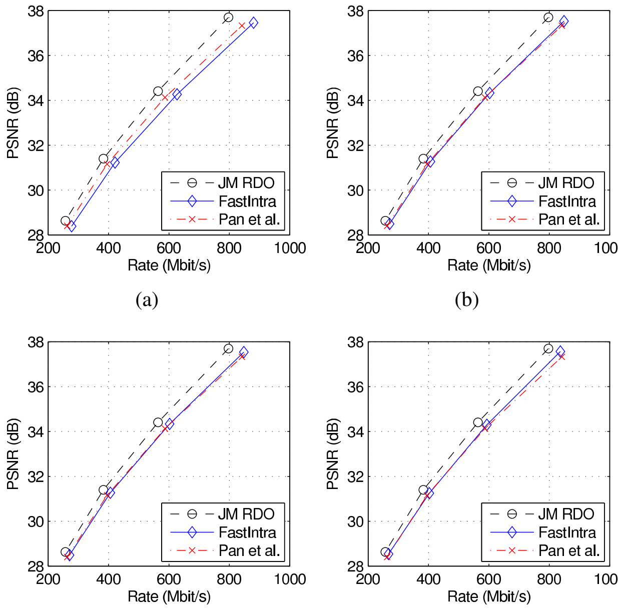


Fig. 7. Block diagram for the general Fast Intra algorithm.



(a)

(b)

(c)

(d)

Fig. 8. PSNR vs. rate for the sequence `news` QCIF coded with all Intra frames, $QP = \{28, 32, 36, 40\}$, and different parameters $\overline{M}$ and T (`Intra8x8` mode disabled). a) $(\overline{M}, T) = (5, 3)$ b) $(\overline{M}, T) = (6, 2)$ c) $(\overline{M}, T) = (6, 3)$ d) $(\overline{M}, T) = (7, 3)$.

with respect to the reference software (with `Intra8x8` mode disabled). The computational saving has been calculated measuring the coding time $T_{JM}$for the standard full-complexity algorithm in JM codec (without `Intra8x8`) and the coding time T_{BP} for the proposed approach. The parameter ΔTime is then equal to

$$\Delta\text{Time} = \frac{T_{BP} - T_{JM}}{T_{JM}} \cdot 100. \qquad (12)$$

The average values of the PSNR loss and the rate increment for each sequence have been obtained using the same approach reported by Pan *et al.* in [3]. For each configuration $(\overline{M}, T)$, Table I also reports the average values E[ΔBits], E[ΔPSNR], E[ΔTime] after the each table entry $(\overline{M}, T)$. These mean performances have been obtained averaging the values of the parameters ΔBits, ΔPSNR, and ΔTime for all the video sequences, respectively. The presented algorithm is able to reduce the coding time of approximately 63% with respect to the JM exhaustive approach with an average rate increment lower than 5% and a PSNR loss of 0.16 dB ($\overline{M} = 6$ and $T = 2$).

It is also possible to notice that the performance of the proposed approach in terms of rate-distortion optimization is slightly better for complex sequences that present many details (compare the results for `coastguard` and `bus` with respect to those for `news` and `tempete`).

The DD algorithm described in Subsection IV-B makes possible to improve the relative coding time reduction of an additional 13% (compare results for $\overline{M} = 6$ and $T = 2$ with results for $\overline{M} = 6$ without DD).

The parameter T also permits a finer tuning of the computational complexity. The results reported in Table I show that reducing the parameter $\overline{M}$ by 1 permits a decrement of the computational complexity between 4.44% and 8.37%. On the other hand, increasing or decreasing the value of T by 1 leads to variation of the ΔTime between 2.21% and 3.67%.

The reported data also show that the rate-distortion performance is slightly better than that of the approach in [3]. Moreover, significant improvements in the stability of the complexity saving are also noticeable with respect to the results of the approach reported in [11]. The bottom part of Table I reports the results for the algorithm proposed by Pan *et al.* in [3]. Equalizing the rate increment, the performance of the proposed algorithm with $\overline{M} = 7$ and $T = 2$ permits reducing the PSNR loss of 0.04 dB and improving the coding time saving of approximately 2%. Despite this slight improvement, the real advantage of the proposed approach relies on the possibility of forecasting the computational complexity required by coding operations. It is possible to notice that the computational complexity does not significantly vary according to the input sequence, since the maximum deviation of time saving from its average with $\overline{M} = 7$ and $T = 2$ is 0.79% while it is equal to 5.81% (with 59.72% average time saving) for the algorithm of Pan *et al.* [3] and 30.38% (with 60.38% average time saving) for the solution proposed by Yong-dong *et al.* [11]. These measurements have been obtained computing the means E[ΔTime] and finding the maximum difference between E[ΔTime] and the savings ΔTime of the coded sequences (see [3], [11]). Moreover, it is possible to tune the parameters of the fast estimation algorithm in order to vary the required computational complexity and the rate-distortion performance.

TABLE I
EXPERIMENTAL RESULTS WITH `INTRA8x8` DISABLED AND ONLY INTRA FRAMES.

$(\overline{\mathbf{M}}, \mathbf{T})$	**Sequence**	**Δ Bits (%)**	**Δ PSNR**	**Δ Time (%)**
$(5, 2)$	container (qcif)	10.42	-0.18	-72.24
	news (qcif)	9.42	-0.23	-72.44
	coastguard (qcif)	8.04	-0.18	-71.55
	bus (cif)	6.22	-0.21	-70.50
	tempete (cif)	4.55	-0.30	-69.42
average E[·]		7.73	-0.22	-71.23
$(6, 2)$	container (qcif)	5.35	-0.14	-62.56
	news (qcif)	5.79	-0.16	-62.46
	coastguard (qcif)	3.55	-0.14	-62.64
	bus (cif)	2.15	-0.16	-63.64
	tempete (cif)	4.95	-0.22	-63.00
average E[·]		4.36	-0.16	-62.86
$(6, 3)$	container (qcif)	5.32	-0.14	-59.95
	news (qcif)	5.68	-0.15	-60.43
	coastguard (qcif)	3.67	-0.14	-60.48
	bus (cif)	2.12	-0.12	-61.32
	tempete (cif)	4.85	-0.20	-61.05
average E[·]		4.33	-0.15	-60.65
$(7, 2)$	container (qcif)	3.77	-0.13	-58.98
	news (qcif)	4.55	-0.16	-57.85
	coastguard (qcif)	2.14	-0.13	-59.21
	bus (cif)	2.11	-0.14	-58.19
	tempete (cif)	4.61	-0.19	-57.88
average E[·]		3.44	-0.15	-58.42
$(7, 3)$	container (qcif)	3.60	-0.12	-55.18
	news (qcif)	4.50	-0.14	-54.14
	coastguard (qcif)	2.19	-0.12	-55.25
	bus (cif)	2.13	-0.13	-54.83
	tempete (cif)	4.57	-0.18	-54.36
average E[·]		3.40	-0.14	-54.75
6 w/o DD	container (qcif)	5.64	-0.08	-50.43
	news (qcif)	4.96	-0.09	-47.79
	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 E[·]		4.40	-0.10	-49.76
Pan *et al.*	container (qcif)	3.69	-0.23	-56.36
	news (qcif)	3.90	-0.29	-55.34
	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 E[·]		3.46	-0.19	-56.51

As a consequence, the computational saving can be parameterized as a function of $\overline{M}$ and T (see Fig. 9) since the relative decrement in the computational time does not depend on the characteristics of the input signal. Experimental results show that it is possible to approximate the complexity saving via the equation

$$\Delta\text{Time} = w_1 M + w_2 T + w_3 M T \qquad (13)$$

where the coefficients w_i, $i = 1, 2, 3$, can be estimated via a multivariate regression. A similar relation can be found for the average PSNR loss and the average rate increment. As a matter of fact, these equations can be included into a rate-distortion optimization problem adding an extra bound to the computational complexity and obtaining the ideal values for $\overline{M}$ and T.

In addition, Table II 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. Note

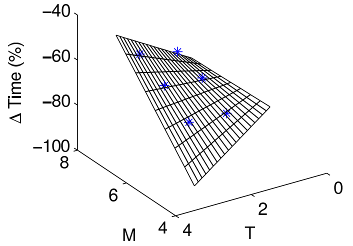


Fig. 9. Average time saving (ΔTime) as function of $\overline{M}$ and T. Points report the experimental values, while the surface show the values estimated via the model in eq. (13).

TABLE II
EXPERIMENTAL RESULTS WITH Intra8x8 ENABLED AND ONLY INTRA FRAMES.

$(\overline{\mathbf{M}},\mathbf{T})$	**Sequence**	**Δ Bits (%)**	**Δ PSNR**	**Δ Time (%)**
$(6,3)$	container (qcif)	5.93	−0.14	−66.97
	news (qcif)	6.40	−0.19	−64.63
	coastguard (qcif)	5.09	−0.18	−66.51
	bus (cif)	3.42	−0.20	−63.69
	tempete (cif)	5.66	−0.22	−61.78
average E[·]		5.30	−0.18	−64.72
$(7,2)$	container (qcif)	4.27	−0.13	−62.76
	news (qcif)	5.34	−0.17	−59.49
	coastguard (qcif)	3.40	−0.16	−62.37
	bus (cif)	3.12	−0.17	−59.73
	tempete (cif)	4.99	−0.20	−58.02
average E[·]		4.23	−0.17	−60.47

also that the computational saving increases since the rate-distortion optimization process becomes more complex as the mode Intra8x8 is added, and therefore, the adoption of fast method for Intra prediction proves to be an effective strategy in the coding process. Table III reports additional results obtained for higher resolution sequences (SDTV 720×480 [18] and 720p 1280×720 [19]). Results were obtained for QP=$\{20, 24, 28, 32, 36, 40\}$ with CABAC entropy coder. For SDTV sequence, the first 100 frames were coded with no skipping, while for 720p sequences 30 pictures were coded skipping 10 frames between two coded pictures. The reported results shows that the stronger spatial correlation due to the higher pixel resolution permits improving the relative time saving.

TABLE III
EXPERIMENTAL RESULTS WITH Intra8x8 ENABLED AND ONLY INTRA FRAMES FOR 480P SDTV AND 720P SEQUENCES .

$(\overline{\mathbf{M}},\mathbf{T})$	**Sequence**	**Δ Bits (%)**	**Δ PSNR**	**Δ Time (%)**
$(6,3)$	whale show	5.24	−0.26	−67.64
	opening ceremony	3.35	−0.25	−66.75
	driving	7.21	−0.27	−66.67
	football	6.50	−0.22	−71.29
	stefan	4.45	−0.20	−68.91
	mobcal (720p)	1.20	−0.48	−66.37
	stockholm (720p)	1.12	−0.41	−66.82
average E[·]		4.15	−0.30	−67.78
$(7,2)$	whale show	4.91	−0.22	−61.65
	opening ceremony	3.30	−0.22	−61.28
	driving	6.78	−0.24	−61.99
	football	5.17	−0.19	−66.21
	stefan	3.53	−0.18	−64.78
	mobcal (720p)	1.82	−0.41	−59.59
	stockholm (720p)	2.35	−0.33	−58.40
average E[·]		3.98	−0.26	−62.19

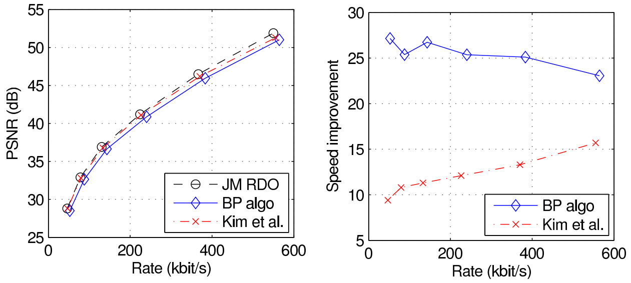
TABLE IV
EXPERIMENTAL RESULTS WITH GOP IP. . .P.

$(\overline{\mathbf{M}},\mathbf{T})$	**Sequence**	**Δ Bits (%)**	**Δ PSNR**	**Δ Time (%)**
$(6,3)$ w/o Intra 8x8	container (qcif)	1.19	−0.04	−25.00
	news (qcif)	1.00	−0.05	−25.70
	coastguard (qcif)	0.03	−0.01	−27.00
	bus (cif)	0.23	−0.01	−26.88
	tempete (cif)	0.44	−0.01	−23.82
average E[·]		0.58	−0.02	−25.68
$(6,3)$ Intra 8x8	container (qcif)	1.25	−0.04	−31.59
	news (qcif)	1.05	−0.07	−30.78
	coastguard (qcif)	0.22	−0.01	−32.10
	bus (cif)	0.25	−0.01	−33.70
	container (qcif)	1.25	−0.04	−31.59
average E[·]		0.80	−0.03	−31.95
Pan *et al.*	container (qcif)	1.80	−0.08	−20.78
	news (qcif)	1.23	−0.07	−23.11
	coastguard (qcif)	0.50	−0.02	−21.20
	bus (cif)	0.32	−0.01	−26.05
	tempete (cif)	0.81	−0.03	−26.72
average E[·]		0.93	−0.04	−23.57

Following the same evaluation procedure of [3], some tests were devoted to evaluate the impact of the BP-based fast Intra prediction on the complexity of the overall coding process. Table IV reports the coding results for GOP of 100 frames with structure IP. . .P where Intra8x8 mode is both enabled and disabled. It is possible to notice that the proposed method improves the results of the algorithm by Pan *et al.* both in terms of rate-distortion performance (it obtained lower rate increment and quality decrement for $\overline{M} = 6$) and of complexity reduction (the proposed approach permits an average 25.88% reduction in the coding time with respect to the 23.57% reduction of the algorithm in [3]).

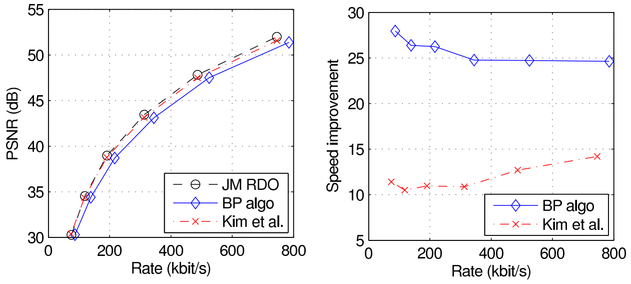
Final tests compare the proposed approach with more recent Fast Intra strategies, like the one proposed by Kim *et al.* [8]. Sequences table and akiyo were coded with $QP = \{10, 16, 22, 28, 34, 40\}$ like in [8]. In this case, the coding speed of the Intra prediction algorithm is improved by an effective Rate-Distortion Optimization (RDO) strategy that proves to be less computationally-demanding than the algorithm proposed within the JVT. As a results, the coding time saving is greater since the BP approach employs the standard RDO algorithm which is computationally expensive. To be fair, experimental tests compared the solution in [8] and the BP approach with RDO disabled. Experimental results in Fig. 10 show that the algorithm performs quite well and permits improving the coding speed by an average factor of 25 (while the approach by Kim *et al.* improves the speed by an average factor of 12).

Similar results have been obtained for the sequence akiyo (see Fig. 11), which presents a higher spatial correlation with respect to the sequence table. In this case, the BP approach produces a higher increment in the coded bit rate since the RDO algorithms in the JVT algorithm and in the approach by Kim *et al.* performs better. From the reported plots it is possible to notice an 8% increase in the coded bit rate is associated to a speed improvement equal to 25. The approach in [8] constrains the bit rate increment to 3%, but the associated speed improvement is only 13 on average. As a matter of fact, the performance of the proposed algorithm still remains competitive with respect to the other approach.



(a) (b)

Fig. 10. Performance of the algorithm with RDO disabled for the sequence table QCIF (with $QP = 10, 16, 22, 28, 34, 40$, frame skip = 5, and Intra only frames). a) PSNR vs. Rate b) Speed improvement vs. rate.



(a) (b)

Fig. 11. Performance of the algorithm with RDO disabled for the sequence akiyo QCIF (with $QP = 10, 16, 22, 28, 34, 40$, frame skip = 5, and Intra only frames). a) PSNR vs. Rate b) Speed improvement vs. rate.

Future work will be focused on designing an effective RDO strategy that requires a lower computational effort with respect to the algorithm proposed within the JVT.

VII. 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 a Belief Propagation strategy that relies on the statistical dependence existing between spatially neighboring blocks. At the same time, 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 62%) 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.

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] S. Cho, Z. Bojković, D. Milovanović, J. Lee, and J.-J. Hwang, "Image quality evaluation: Jpeg 2000 versus intra-only h.264/avc high profile," *Facta universitatis - series: Electronics and Energetics*, vol. 20, no. 1, pp. 71 – 84, Apr. 2007.

[3] 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.

[4] F. Pan, X. Lin, S. Rahardja, K. P. Lim, and Z. G. Li, "A directional field based fast Intra mode decision algorithm for H.264 video coding," in *Proc. of IEEE ICME 2004*, Taipei, Taiwan, June 27 – 30, 2004, pp. 1147–1150.

[5] J.-S. Ryu and E.-T. Kim, "Fast Intra coding method of H.264 for video surveillance system," *IJCSNS*, vol. 7, no. 10, pp. 76–81, Oct. 2007.

[6] 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.

[7] J. Jeong and D. N. Kwon, "DCT Based Fast 4X4 Intra-Prediction Mode Selection," in *Proc. of IEEE CCNC 2007*, Las Vegas, NV, Jan. 11 – 13, 2007, pp. 332 – 335.

[8] C. Kim, H.-H. Shih, and C.-C. Jay Kuo, "Fast H.264 intra-prediction mode selection using joint spatial and transform domain features," *J. Vis. Commun. Image R.*, vol. 17, no. 2, pp. 291–310, Apr. 2006.

[9] 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.

[10] F. Lorás and J.-C. Amiel, "Method for finding the prediction direction in intraframe video coding," International Patent WO 2005/088979, Jan. 2005.

[11] 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.

[12] H. Kalva and L. Christodoulou, "Using Machine Learning for Fast Intra MB Coding in H.264," in *Proc. of the SPIE-VCIP 2007*, San Jose, CA, USA, Feb. 2007, pp. 65082U–1 – 65082U–4.

[13] 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.

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

[15] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publisher, Norwell, MA, USA, 1991.

[16] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, July 2003.

[17] J. Bialkowski, A. Kaup, and K. Illgner, "Fast transcoding of intra frames between H.263 and H.264," in *Proc. of IEEE ICIP 2004*, Singapore, Oct. 24 – 27, 2004, pp. 1151–1154.

[18] "Stanford Center for Image Systems Engineering: Test Images and Videos," 2009, Available at: http://scien.stanford.edu/labsite/scien_test_images_videos.php.

[19] "Xiph.org Test Media," 2009, Available at: http://media.xiph.org.

Simone Milani was born in Camposampiero (PD), Italy, in 1978. From the University of Padova, Italy, he received the Laurea degree in Telecommunication Engineering in 2002, and the Ph.D. degree in Electronics and Telecommunication Engineering in 2007. In 2006 he was a visiting Ph.D. student at the University of California-Berkeley under the supervision of prof. K. Ramchandran, while in 2007 he was a post-doc researcher at the University of Udine, Italy, collaborating with prof. R. Rinaldo. At the moment he is post-doc researcher at the Department of Information Engineering at the University of Padova working with prof. G. Calvagno. He has also worked with STMicroelectronics, Agrate Brianza, Italy. His main research topics are digital signal processing, source coding, joint source-channel coding, robust video transmission over lossy packet networks, distributed source coding.